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
import multiprocessing as mp
import pickle
import random

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)
```

```
2024-06-20 04:32:54.072125: I tensorflow/core/util/port.cc:113] oneDNN custom operations are
2024-06-20 04:32:54.635215: I tensorflow/core/platform/cpu_feature_guard.cc:210] This Tensorf
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebu
```

Model itself

```
np.random.seed(590154)

population = 10000
initial_infecteds = 100
epidemic_length = 1000  # not used
number_of_events = 200000

pv_champ_alpha = 0.95 * 0.13  # prop of effective care
pv_champ_beta = 0.429  # prop of radical cure
pv_champ_gamma_L = 1 / 383  # liver stage clearance rate
pv_champ_delta = 0.05  # prop of imported cases
pv_champ_lambda = 0.01  # transmission rate
pv_champ_f = 1 / 69  # relapse frequency
pv_champ_r = 1 / 60  # blood stage clearance rate

gamma_L_max = 1 / 30
lambda_max = 0.05
f_max = 1 / 20
r_max = 1 / 15

upper_bounds = np.array([1, 1, gamma_L_max, lambda_max, f_max, r_max])

num_lhc_samples = 50
initial_repeats = 1
dis_mean_n = 30
```

```
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,
    seed=12,
):
    np.random.seed(seed)
    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
        + delta_t * alpha_ * (1 - beta_) * (f + lambda_ * (I_L + I_0) / N) * S_L
    )
```

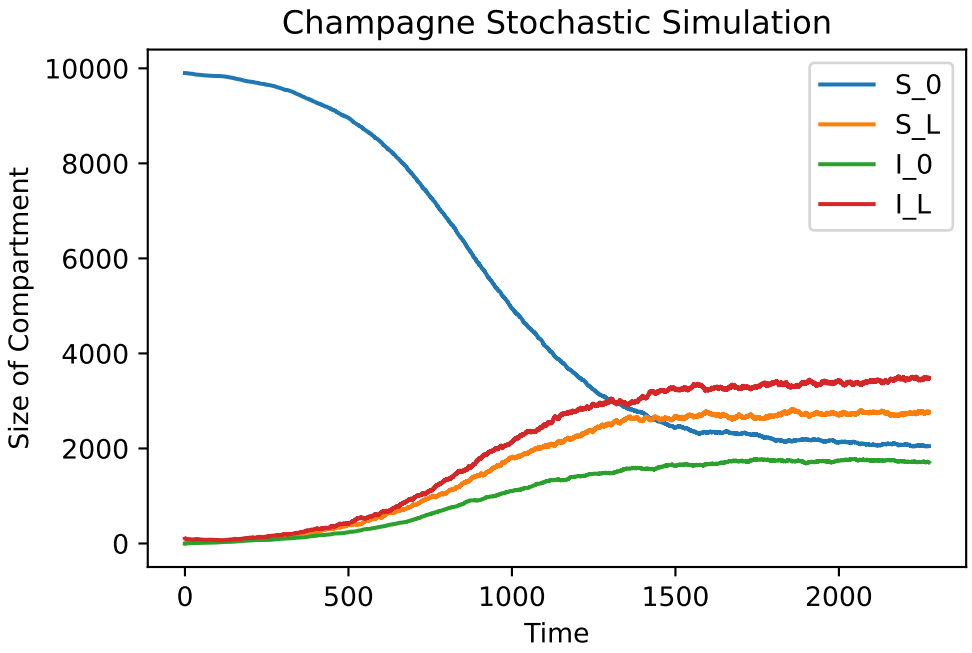
```
        new_stages = np.random.choice(
            [
                {
                    "t": t,
                    "S_0": S_0 - 1,
                    "S_L": S_L,
                    "I_0": I_0,
                    "I_L": I_L + 1,
                    "inc_counter": inc_counter + silent_incidences + 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,
)
```

Plotting outcome

```
champ_samp.drop("inc_counter", axis=1).plot(x="t", legend=True)
plt.xlabel("Time")
plt.ylabel("Size of Compartment")
plt.title("Champagne Stochastic Simulation")
plt.savefig("champagne_GP_images/champagne_simulation.pdf", bbox_inches='tight')
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, seed=12301923):
    champ_df_ = champagne_stochastic(alpha_, beta_, gamma_L, lambda_, f, r, seed=seed)
    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)

    first_month_ind = champ_df_[champ_df_["t"].le(30)].index[-1]
    first_month_prev = (
        champ_df_.iloc[first_month_ind]["I_0"]
        + champ_df_.iloc[first_month_ind]["I_L"]
    )
    fin_prev = champ_df_.iloc[-1]["I_0"] + champ_df_.iloc[-1]["I_L"]

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


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,
)

print(observed_sum_stats)


def single_discrepency(alpha_, beta_, gamma_L, lambda_, f, r, seed=12301923):
    x = champ_sum_stats(alpha_, beta_, gamma_L, lambda_, f, r, seed=seed)
    return np.log(np.linalg.norm((x - observed_sum_stats) / observed_sum_stats))
```

```
def discrepency_fn(alpha_, beta_, gamma_L, lambda_, f, r, mean_of=dis_mean_n):
    seed = int(np.random.uniform() * 1000000)
    with mp.Pool(processes=mp.cpu_count()) as pool:
        args = [
            (alpha_, beta_, gamma_L, lambda_, f, r, seed * i) for i in range(mean_of)
        ]
        results = pool.starmap(single_discrepency, args)
    mean_obs = np.mean(results)
    return mean_obs
```

```
[ 461. 5205.   42.   87.]
```

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 = 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  0.100008  0.122349  0.009668  0.015376  0.016920  0.015954
1  0.659225  0.590955  0.001070  0.038947  0.007433  0.003318
2  0.503558  0.005003  0.031832  0.027053  0.002028  0.019736
3  0.011840  0.630562  0.023631  0.033488  0.035622  0.035127
4  0.271011  0.942434  0.014052  0.030138  0.031133  0.051736
```

Generate Discrepencies

```
LHC_samples_reps = np.repeat(LHC_samples, dis_mean_n, axis=0)

with mp.Pool(processes=mp.cpu_count()) as pool:
    args = [
        (a, b, c, d, e, f, int(g * np.random.uniform()))
        for (a, b, c, d, e, f), g in zip(
            list(map(tuple, LHC_samples_reps)), range(LHC_samples_reps.shape[0])
        )
    ]
    results = pool.starmap(single_discrepency, args)

random_discrepencies = np.mean(np.array(results).reshape(-1, dis_mean_n), axis=1)

print(random_discrepencies)
```

```
[-0.72112073  0.85173495 -0.10445358  1.04268897  0.56652755  0.46286618
  0.36628974  0.78458654  0.64547033 -0.43466706  0.72250049 -0.10120307
 -0.0321187  -0.16726528  1.26145031  0.92538462 -0.26121501  0.3277712
 -0.70371735  0.2155892   0.81173955  0.59710669  0.51707793  1.33971327
  1.0812211   0.8602283   1.38585448 -0.09602358  0.61782169  0.42402107
  0.31068428 -0.04423537 -0.96457358  0.38741823  0.24964537  0.14481727
  0.22145299  0.33272269 -0.67052626  1.12934722  1.21789051  0.20304941
  0.39821991  0.35048571  0.18218027  1.24579385 -0.49153258  0.34009135
  1.04344262 -0.03480152]
```

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())
```

```
2024-06-20 04:37:18.605101: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1928] Created
2024-06-20 04:37:18.605706: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1928] Created
2024-06-20 04:37:18.606125: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1928] Created
2024-06-20 04:37:18.606492: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1928] Created
```

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

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 / 2, 1.0 / 2, gamma_L_max / 2, lambda_max / 2, f_max / 2, r_max / 2],
    # ),
    bijector=constrain_positive,
    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
array([-2.07944154, -2.07944154, -5.48063892, -5.07517382, -5.07517382,
       -4.78749174])>, <tf.Variable 'observation_noise_variance_champ:0' shape=() dtype=float
```

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
```

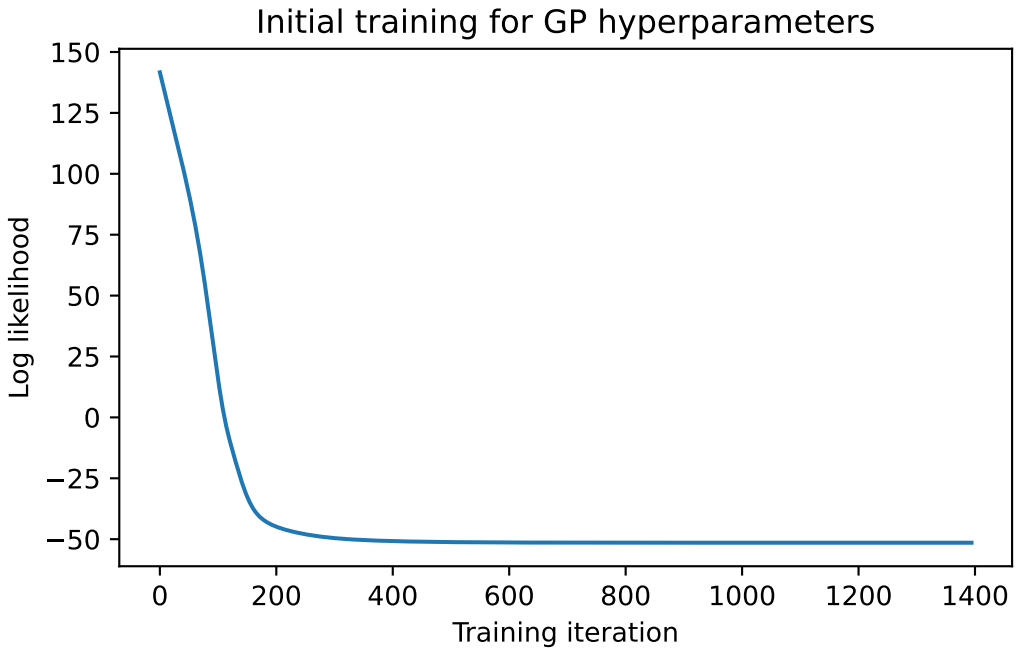
```
2024-06-20 04:37:23.221628: I tensorflow/core/util/cuda_solvers.cc:178] Creating GpuSolver ha
```

```
Hyperparameter convergence reached at iteration 1395.
```

```
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 / 2,
        #                     1.0 / 2,
        #                     gamma_L_max / 2,
        #                     lambda_max / 2,
        #                     f_max / 2,
        #                     r_max / 2,
        #                 ],
        #             )
        #             .forward(var)
        #             .numpy()
        #             .round(3),
        #         )
        #     )
        # else:
        print(
            "{} is {}\n".format(
                var.name, constrain_positive.forward(var).numpy().round(3)
            )
        )

initial_losses_LOOCV = lls_
```

```
Trained parameters:
amplitude_champ:0 is 1.526

length_scales_champ:0 is [0.914 3.298 0.076 0.014 0.061 0.075]

observation_noise_variance_champ:0 is 0.018

bias_mean:0 is 0.898
```

```
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", bbox_inches="tight")
plt.show()
```



Creating slices across one variable dimension

```
plot_samp_no = 21
plot_samp_times = 10
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,
        ),
        plot_samp_times,
        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,
    ),
    plot_samp_times,
    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
            ),  # 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,
    ),
    plot_samp_times,
    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
            ),  # 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,
    ),
    "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
                ),  # 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,
        ),
        plot_samp_times,
        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,
        ),
        plot_samp_times,
        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
            ),
```

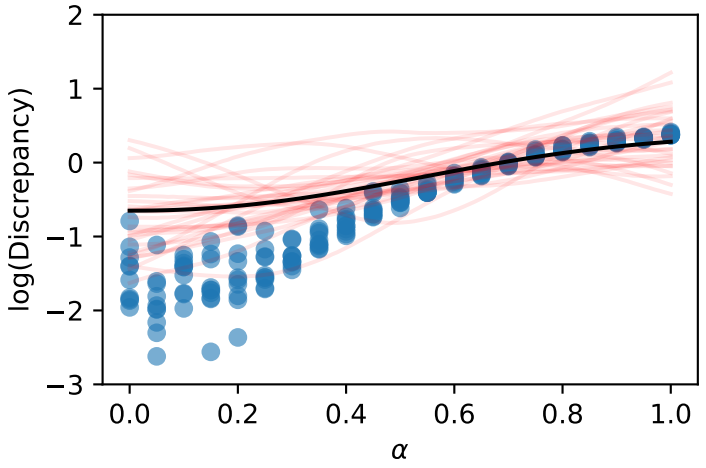
```
            axis=1,
        ),
        plot_samp_times,
        axis=0,
    ),
    "r_gp_samples": np.concatenate(
        (
            np.repeat(pv_champ_alpha, plot_gp_no).reshape(-1, 1),  # alpha
            np.repeat(pv_champ_beta, plot_gp_no).reshape(-1, 1),  # beta
            np.repeat(pv_champ_gamma_L, 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,
    ),
}
```

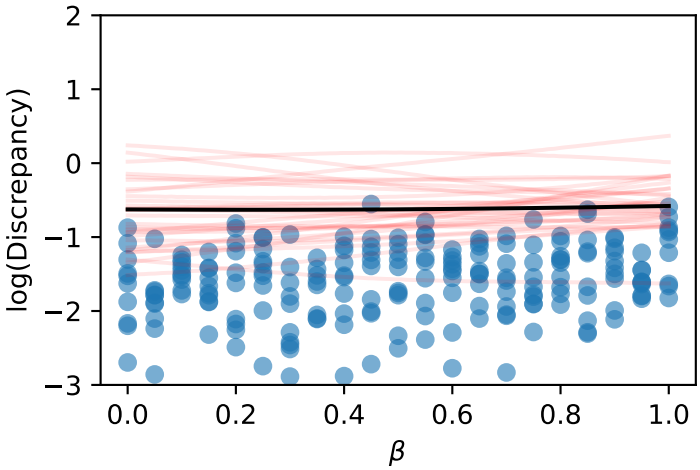
Plotting the GPs across different slices

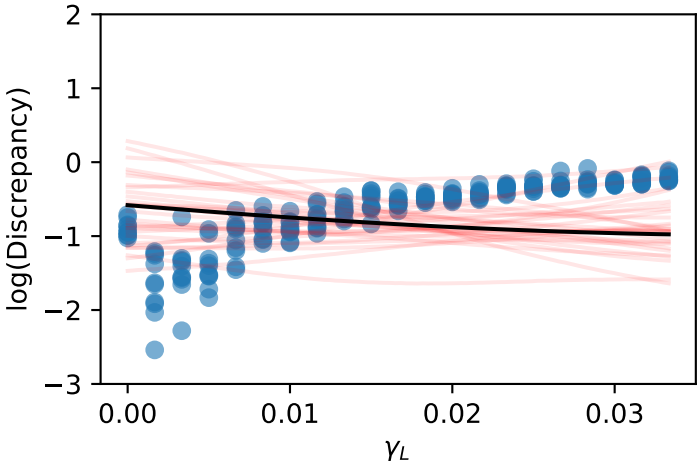
```
GP_seed = tfp.random.sanitize_seed(4362)
vars = ["alpha", "beta", "gamma_L", "lambda", "f", "r"]
slice_indices_dfs_dict = {}
slice_index_vals_dict = {}
slice_discrepencies_dict = {}
gp_samples_dict = {}

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

    df_temp = val_df.assign(seed=range(val_df.shape[0]))
    seed = int(np.random.uniform() * 1000000)
    with mp.Pool(processes=mp.cpu_count()) as pool:
        args = list(df_temp.itertuples(index=False, name=None))
        results = pool.starmap(single_discrepency, args)
```

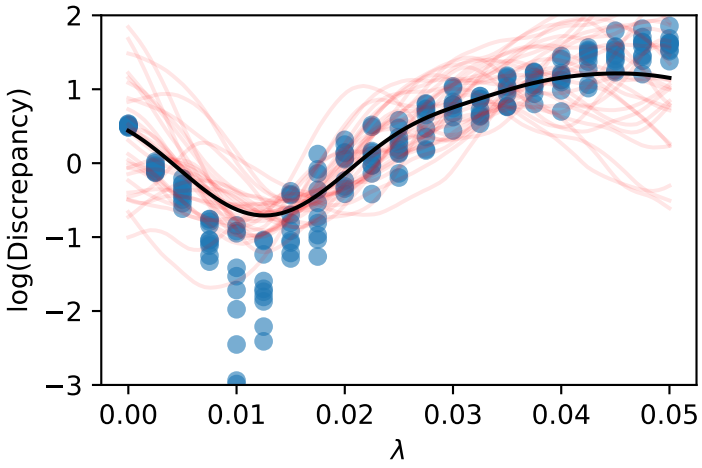
```
discreps = results
slice_discrepencies_dict[var + "_slice_discrepencies"] = discreps

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)

gp_samples_dict[var + "initial_gp_samps"] = GP_samples

plt.figure(figsize=(4, 2.5))
plt.scatter(
    val_df[var].values,
    discreps,
    label="Untrained Discrepencies",
    alpha=0.6,
)
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
```

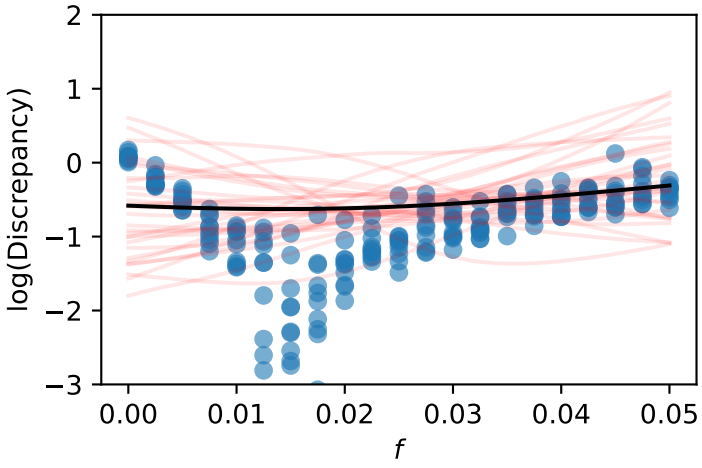
```
    ),
    c="black",
    alpha=1,
    label="Posterior Mean",
)
# leg = plt.legend(loc="upper left")
# for lh in leg.legend_handles:
#     lh.set_alpha(1)
if var in ["f", "r"]:
    plt.xlabel("$" + var + "$")
    # plt.title("$" + var + "$ slice before Bayesian Acquisition")
else:
    plt.xlabel("$\\" + var + "$")
    # 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, 2))
plt.savefig(
    "champagne_GP_images/initial_" + var + "_slice_log_discrep.pdf",
    bbox_inches="tight",
)
plt.show()
```

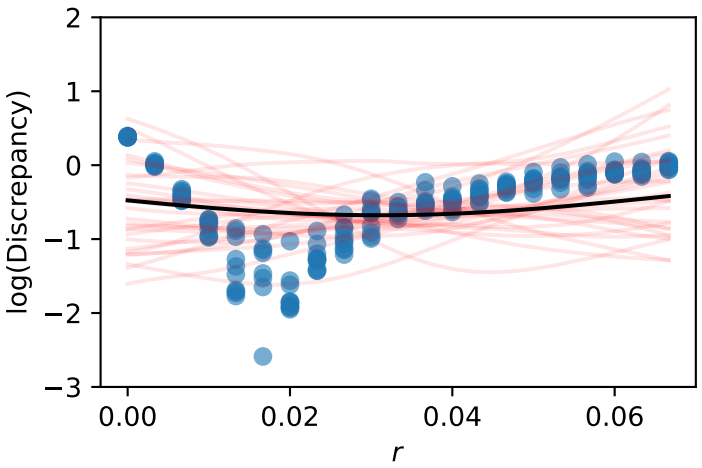



log(Discrepancy)
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
β


log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
γ_L


log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
λ


log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
f



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 2.328
Others Kernel is [0.096 0.221 0.217 0.336 0.441 0.556 0.074 0.265 0.167 0.091 0.464 0.044
 0.085 0.245 0.529 0.578 0.149 0.712 0.066 0.313 0.369 0.221 0.614 0.368
 0.224 0.297 0.495 0.145 0.185 0.02  0.202 0.072 0.103 0.036 0.056 0.089
 0.043 0.093 0.086 0.576 0.613 0.032 0.192 0.139 0.061 0.573 0.066 0.247
 0.216 0.189]
[[ 3.74331623e+00 -1.79222250e-02 -4.92029511e-02 ... -2.68667304e-02
   3.64973536e-03 -1.27791184e+00]
 [-1.79222250e-02  2.81344520e+00  1.52743886e-01 ...  3.71902870e-01
  -1.15218941e+00 -1.96118052e-01]
 [-4.92029511e-02  1.52743886e-01  3.32181636e+00 ... -1.19911743e+00
  -1.01439904e-02 -1.78146988e-01]
 ...
 [-2.68667304e-02  3.71902870e-01 -1.19911743e+00 ...  4.68435369e+00
  -6.18945598e-02  1.72952538e-01]
 [ 3.64973536e-03 -1.15218941e+00 -1.01439904e-02 ... -6.18945598e-02
   2.63445217e+00  3.72076630e-02]
 [-1.27791184e+00 -1.96118052e-01 -1.78146988e-01 ...  1.72952538e-01
   3.72076630e-02  4.03959172e+00]]
Variance function is [2.025]
Variance function is 2.007
```

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
```

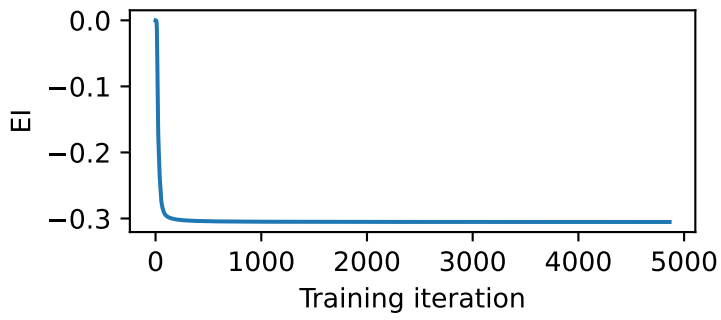
```
(<tf.Variable 'next_alpha:0' shape=() dtype=float64, numpy=0.0>,
 <tf.Variable 'next_beta:0' shape=() dtype=float64, numpy=0.0>,
 <tf.Variable 'next_gamma_L:0' shape=() dtype=float64, numpy=0.0>,
 <tf.Variable 'next_lambda:0' shape=() dtype=float64, numpy=0.0>,
 <tf.Variable 'next_f:0' shape=() dtype=float64,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

```
curr_min = min(champ_GP_reg.mean_fn(index_vals))

def EI_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 = champ_GP_reg.stddev(index_points=next_guess)
    delt = curr_min - mean_t
    return -tf.squeeze(
        delt * tfd.Normal(0, np.float64(1)).cdf(delt / std_t)
        + std_t * tfd.Normal(0, np.float64(1)).prob(delt / std_t)
    )

first_optimizer = tf.keras.optimizers.Adam(learning_rate=0.05)

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

num_iters = 10000

lls_ = np.zeros(num_iters, np.float64)
tolerance = 1e-8  # 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)
    )
)
```

```
Acquisition function convergence reached at iteration 4864.
The final EI loss was -0.305 with predicted mean of [-1.043]
```

```
plt.figure(figsize=(3.8, 1.5))
plt.plot(lls_)
plt.xlabel("Training iteration")
plt.ylabel("EI")
plt.savefig("champagne_GP_images/initial_EI_loss_training.pdf", bbox_inches="tight")
plt.show()
```



```
def update_GP_LOO(champ_GP, index_vals, obs_vals, observation_noise_variance_champ):

    def LOO_loss(champ_GP, index_vals, obs_vals, observation_noise_variance_champ):
        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
        return -tf.math.reduce_sum(log_var - log_mu)

    @tf.function(autograph=False, jit_compile=False)
    def opt_GP():
        with tf.GradientTape() as tape:
            loss = LOO_loss(
                champ_GP, index_vals, obs_vals, observation_noise_variance_champ
            )
        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 - 0.1
        return -tf.squeeze(
            delt * tfd.Normal(0, np.float64(1)).cdf(delt / std_t)
            + std_t * tfd.Normal(0, np.float64(1)).prob(delt / std_t)
        )

    optimizer_fast = tf.keras.optimizers.Adam(learning_rate=0.05)

    @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-8  # 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)
        )
    )


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()
```

```
num_slice_updates = 15

all_slices = [
    np.linspace(0, 1, num_slice_updates + 2, dtype=np.float64)[1:-1],  # alpha
    np.linspace(0, 1, num_slice_updates + 2, dtype=np.float64)[1:-1],  # beta
    np.linspace(0, gamma_L_max, num_slice_updates + 2, dtype=np.float64)[
        1:-1
    ],  # gamma_L
    np.linspace(0, lambda_max, num_slice_updates + 2, dtype=np.float64)[1:-1],  # lambda
    np.linspace(0, f_max, num_slice_updates + 2, dtype=np.float64)[1:-1],  # f
    np.linspace(0, r_max, num_slice_updates + 2, dtype=np.float64)[1:-1],  # r
]
```

```
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())
    #     )

    index_update = 0
    for var in [next_alpha, next_beta, next_gamma_L, next_lambda, next_f, next_r]:
        if np.random.uniform() > 0.5:
            var.assign(min_index[index_update])
        else:
            var.assign(
                var.bijector.forward(np.float64(100000000.0))
                * np.float64(np.random.uniform())
            )
        index_update += 1

    # 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,
    )

    bayes_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(bayes_params.round(3)))

    if t < 6:
        new_params = np.repeat(
            np.array(
                [
                    [
                        next_alpha.numpy(),
                        next_beta.numpy(),
                        next_gamma_L.numpy(),
                        next_lambda.numpy(),
                        next_f.numpy(),
                        next_r.numpy(),
                    ]
                ]
            ),
            num_slice_updates,
            axis=0,
        )
        new_params[:, t % 6] = all_slices[t % 6]
    else:
        new_params = np.repeat(
```

```
            np.array(
                [
                    [
                        next_alpha.numpy(),
                        next_beta.numpy(),
                        next_gamma_L.numpy(),
                        next_lambda.numpy(),
                        next_f.numpy(),
                        next_r.numpy(),
                    ]
                ]
            ),
            4,
            axis=0,
        )
        new_params[:, t % 6] = np.random.uniform(0, upper_bounds[t % 6], 4)

    new_params = np.append(
        new_params,
        np.array(
            [
                [
                    next_alpha.numpy(),
                    next_beta.numpy(),
                    next_gamma_L.numpy(),
                    next_lambda.numpy(),
                    next_f.numpy(),
                    next_r.numpy(),
                ]
            ]
        ),
        axis=0,
    )

    new_params_reps = np.repeat(new_params, dis_mean_n, axis=0)
    index_vals = np.append(index_vals, new_params, axis=0)

    with mp.Pool(processes=mp.cpu_count()) as pool:
        args = [
            (a, b, c, d, e, f, int(g * np.random.uniform()))
            for (a, b, c, d, e, f), g in zip(
```

```
            list(map(tuple, new_params_reps)), range(new_params_reps.shape[0])
        )
    ]
    results = pool.starmap(single_discrepency, args)

new_discrepencies = np.mean(np.array(results).reshape(-1, dis_mean_n), axis=1)

print("The mean of the samples was {}".format(new_discrepencies[-1].round(3)))
obs_vals = np.append(obs_vals, new_discrepencies)

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

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 or t==6:
    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(champ_GP, index_vals, obs_vals, observation_noise_variance_champ)
    # 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 / 2,
        #                     1.0 / 2,
        #                     gamma_L_max / 2,
        #                     lambda_max / 2,
        #                     f_max / 2,
        #                     r_max / 2,
        #                 ],
        #             )
        #             .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)
gp_samples_dict[var + "_gp_samps" + str(t) + "iters"] = GP_samples

plt.figure(figsize=(4, 2.5))
plt.scatter(
    slice_indices_dfs_dict[var + "_slice_indices_df"][var].values,
    slice_discrepencies_dict[var + "_slice_discrepencies"],
    label="Untrained Discrepencies",
    alpha=0.6,
)
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("$" + var + "$")
    # plt.title(
    #     "$" + var + "$ slice after " + str(t) + " Bayesian acquisitions"
    # )
else:
    plt.xlabel("$\\" + var + "$")
    # plt.title(
    #     "$\\" + var + "$ slice after " + str(t) + " Bayesian acquisitions"
    # )
```

```
        plt.ylabel("log(Discrepancy)")
        plt.ylim((-3, 2))
        plt.savefig(
            "champagne_GP_images/"
            + var
            + "_slice_"
            + str(t)
            + "_bolfi_updates_log_discrep.pdf",
            bbox_inches="tight",
        )
        plt.show()

    epsilon = -3

    likelihood_dict = {}
    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)
        likelihood_dict[var + "_slice_means"] = mean
        variance = dis_mean_n * observation_noise_variance_champ.numpy()
        post_std = np.sqrt(variance)
        log_cdf_vals = tfd.Normal(mean, post_std).log_cdf(epsilon)
        likelihood_dict[var + "_synth_log_lik"] = log_cdf_vals

        plt.figure(figsize=(4, 2.5))
        plt.plot(
            slice_indices_dfs_dict[var + "_gp_indices_df"][var].values,
            np.exp(log_cdf_vals),
        )
        if var in ["f", "r"]:
```

```
                plt.xlabel("$" + var + "$")
                # plt.title("Final Synthetic Likelihood for $" + var + "$ Slice")
            else:
                plt.xlabel("$\\" + var + "$")
                # plt.title("Final Synthetic Likelihood for $\\" + var + "$ Slice")
            plt.ylabel("Synthetic likelihood")
            plt.ylim((0, 0.1))
            plt.savefig(
                "champagne_GP_images/" + var + "_slice_" + str(t) + "_synth_likelihood.pdf
                bbox_inches="tight",
            )
            plt.show()
        objects_to_preserve = [
            index_vals,
            obs_vals,
            champ_samp,
            initial_losses_LOOCV,
            slice_samples_dict,
            slice_discrepencies_dict,
            LHC_indices_df,
            gp_samples_dict,
            likelihood_dict,
        ]

        with open("gp_objs.pkl", "wb") as fp:
            pickle.dump(objects_to_preserve, fp)
            print("dictionary saved successfully to file")
```

```
Iteration 0
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.103]
The next parameters to simulate from are [[0.169 0.953 0.018 0.035 0.045 0.02 ]]
The mean of the samples was 1.046
Hyperparameter convergence reached at iteration 5306.
The minimum predicted mean of the observed indices is -0.945 at the point
[0.169 0.759 0.018 0.012 0.026 0.04 ]
Iteration 1
Acquisition function convergence reached at iteration 1649.
The final EI loss was -0.014 with predicted mean of [0.045]
The next parameters to simulate from are [[0.    1.    0.033 0.    0.05  0.067]]
The mean of the samples was 0.385
Iteration 2
```

```
Acquisition function convergence reached at iteration 3902.
The final EI loss was -0.3 with predicted mean of [-0.949]
The next parameters to simulate from are [[0.003 0.    0.033 0.013 0.    0.058]]
The mean of the samples was -0.204
Iteration 3
Acquisition function convergence reached at iteration 2934.
The final EI loss was -0.325 with predicted mean of [-1.239]
The next parameters to simulate from are [[0.258 0.934 0.033 0.013 0.019 0.012]]
The mean of the samples was -0.211
Iteration 4
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 [[0.169 0.759 0.009 0.039 0.026 0.04 ]]
The mean of the samples was 1.097
Iteration 5
Acquisition function convergence reached at iteration 1157.
The final EI loss was -0.26 with predicted mean of [-1.023]
The next parameters to simulate from are [[0.156 0.001 0.033 0.012 0.05  0.048]]
The mean of the samples was -0.748
Iteration 6
Acquisition function convergence reached at iteration 1968.
The final EI loss was -0.103 with predicted mean of [-0.881]
The next parameters to simulate from are [[0.345 0.    0.033 0.013 0.023 0.041]]
The mean of the samples was -0.977
Hyperparameter convergence reached at iteration 4791.
The minimum predicted mean of the observed indices is -1.016 at the point
[0.289 0.    0.033 0.013 0.023 0.041]
Iteration 7
Acquisition function convergence reached at iteration 905.
The final EI loss was -0.009 with predicted mean of [0.243]
The next parameters to simulate from are [[0.453 0.    0.    0.026 0.    0.067]]
The mean of the samples was 0.481
Iteration 8
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.693]
The next parameters to simulate from are [[0.289 0.761 0.018 0.038 0.019 0.057]]
The mean of the samples was 0.628
Iteration 9
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.432]
The next parameters to simulate from are [[0.289 0.    0.018 0.048 0.023 0.041]]
The mean of the samples was 1.368
Iteration 10
```

```
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.249]
The next parameters to simulate from are [[0.12  0.749 0.021 0.04  0.023 0.018]]
The mean of the samples was 1.269
Iteration 11
WARNING:tensorflow:5 out of the last 912 calls to <function update_var_EI.<locals>.opt_var at
Acquisition function convergence reached at iteration 2146.
The final EI loss was -0.036 with predicted mean of [-0.678]
The next parameters to simulate from are [[0.512 0.    0.033 0.012 0.05  0.046]]
The mean of the samples was -0.352
Iteration 12
Acquisition function convergence reached at iteration 2242.
The final EI loss was -0.05 with predicted mean of [-0.702]
The next parameters to simulate from are [[0.603 0.    0.033 0.015 0.    0.023]]
The mean of the samples was -0.619
Iteration 13
Acquisition function convergence reached at iteration 167.
The final EI loss was -0.047 with predicted mean of [-1.008]
The next parameters to simulate from are [[0.291 0.    0.022 0.014 0.024 0.033]]
The mean of the samples was -1.301
Iteration 14
Acquisition function convergence reached at iteration 95.
The final EI loss was -0.02 with predicted mean of [-1.073]
The next parameters to simulate from are [[0.383 0.    0.02  0.014 0.019 0.024]]
The mean of the samples was -1.332
Iteration 15
Acquisition function convergence reached at iteration 85.
The final EI loss was -0.013 with predicted mean of [-1.204]
The next parameters to simulate from are [[0.416 0.    0.024 0.015 0.027 0.026]]
The mean of the samples was -1.053
Iteration 16
Acquisition function convergence reached at iteration 586.
The final EI loss was -0.014 with predicted mean of [-0.182]
The next parameters to simulate from are [[1.    0.001 0.033 0.013 0.05  0.   ]]
The mean of the samples was 0.389
Iteration 17
Acquisition function convergence reached at iteration 105.
The final EI loss was -0.02 with predicted mean of [-1.201]
The next parameters to simulate from are [[0.317 0.    0.022 0.013 0.011 0.029]]
The mean of the samples was -0.924
Iteration 18
Acquisition function convergence reached at iteration 118.
The final EI loss was -0.016 with predicted mean of [-1.132]
```

```
The next parameters to simulate from are [[0.256 0.    0.019 0.013 0.032 0.035]]
The mean of the samples was -0.937
Iteration 19
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.608]
The next parameters to simulate from are [[0.383 0.94  0.011 0.035 0.019 0.024]]
The mean of the samples was 0.644
Iteration 20
Acquisition function convergence reached at iteration 109.
The final EI loss was -0.014 with predicted mean of [-1.162]
The next parameters to simulate from are [[0.411 0.    0.025 0.014 0.018 0.03 ]]
The mean of the samples was -1.2
Hyperparameter convergence reached at iteration 4621.
The minimum predicted mean of the observed indices is -1.273 at the point
[0.383 0.    0.02  0.014 0.019 0.024]
Iteration 21
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.292]
The next parameters to simulate from are [[0.384 0.125 0.02  0.042 0.019 0.014]]
The mean of the samples was 1.271
Iteration 22
Acquisition function convergence reached at iteration 82.
The final EI loss was -0.025 with predicted mean of [-0.934]
The next parameters to simulate from are [[0.499 0.596 0.024 0.014 0.017 0.024]]
The mean of the samples was -0.876
Iteration 23
Acquisition function convergence reached at iteration 1468.
The final EI loss was -0.049 with predicted mean of [-1.237]
The next parameters to simulate from are [[0.243 0.001 0.024 0.013 0.021 0.028]]
The mean of the samples was -1.016
Iteration 24
Acquisition function convergence reached at iteration 107.
The final EI loss was -0.03 with predicted mean of [-1.019]
The next parameters to simulate from are [[0.472 0.    0.017 0.014 0.023 0.018]]
The mean of the samples was -0.933
Iteration 25
Acquisition function convergence reached at iteration 318.
The final EI loss was -0.02 with predicted mean of [-0.341]
The next parameters to simulate from are [[0.555 0.    0.002 0.015 0.009 0.003]]
The mean of the samples was -0.313
Iteration 26
Acquisition function convergence reached at iteration 1307.
The final EI loss was -0.026 with predicted mean of [-0.84]
```

```
The next parameters to simulate from are [[0.29  0.    0.033 0.015 0.012 0.036]]
The mean of the samples was -1.098
Iteration 27
Acquisition function convergence reached at iteration 1637.
The final EI loss was -0.045 with predicted mean of [-0.847]
The next parameters to simulate from are [[0.415 0.863 0.033 0.012 0.037 0.031]]
The mean of the samples was -0.901
Iteration 28
Acquisition function convergence reached at iteration 417.
The final EI loss was -0.032 with predicted mean of [-0.958]
The next parameters to simulate from are [[0.528 0.    0.033 0.013 0.022 0.026]]
The mean of the samples was -1.161
Iteration 29
Acquisition function convergence reached at iteration 118.
The final EI loss was -0.035 with predicted mean of [-1.24]
The next parameters to simulate from are [[0.514 0.    0.023 0.013 0.021 0.024]]
The mean of the samples was -1.264
Iteration 30
Acquisition function convergence reached at iteration 1511.
The final EI loss was -0.023 with predicted mean of [-0.525]
The next parameters to simulate from are [[0.281 0.    0.033 0.009 0.    0.037]]
The mean of the samples was -0.133
Iteration 31
Acquisition function convergence reached at iteration 156.
The final EI loss was -0.027 with predicted mean of [-1.148]
The next parameters to simulate from are [[0.515 0.    0.023 0.016 0.018 0.022]]
The mean of the samples was -1.114
Iteration 32
Acquisition function convergence reached at iteration 1312.
The final EI loss was -0.035 with predicted mean of [-1.16]
The next parameters to simulate from are [[0.346 0.    0.033 0.014 0.022 0.032]]
The mean of the samples was -1.114
Iteration 33
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.513]
The next parameters to simulate from are [[0.383 0.286 0.018 0.03  0.018 0.024]]
The mean of the samples was 0.462
Iteration 34
Acquisition function convergence reached at iteration 588.
The final EI loss was -0.025 with predicted mean of [-0.807]
The next parameters to simulate from are [[0.023 0.    0.017 0.015 0.016 0.034]]
The mean of the samples was -1.259
Iteration 35
```

```
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.164]
The next parameters to simulate from are [[0.385 0.673 0.017 0.045 0.039 0.023]]
The mean of the samples was 1.33
Iteration 36
Acquisition function convergence reached at iteration 90.
The final EI loss was -0.02 with predicted mean of [-0.991]
The next parameters to simulate from are [[0.608 0.    0.027 0.013 0.029 0.023]]
The mean of the samples was -1.141
Iteration 37
Acquisition function convergence reached at iteration 1082.
The final EI loss was -0.024 with predicted mean of [-0.698]
The next parameters to simulate from are [[0.225 0.    0.033 0.013 0.035 0.049]]
The mean of the samples was -0.834
Iteration 38
Acquisition function convergence reached at iteration 1458.
The final EI loss was -0.011 with predicted mean of [-0.183]
The next parameters to simulate from are [[0.881 0.001 0.033 0.006 0.05  0.027]]
The mean of the samples was -0.221
Iteration 39
Acquisition function convergence reached at iteration 1476.
The final EI loss was -0.008 with predicted mean of [0.139]
The next parameters to simulate from are [[1.   0.   0.   0.05 0.05 0.  ]]
The mean of the samples was 1.78
Iteration 40
Acquisition function convergence reached at iteration 1551.
The final EI loss was -0.012 with predicted mean of [-0.007]
The next parameters to simulate from are [[0.999 0.999 0.033 0.017 0.05  0.067]]
The mean of the samples was 0.385
Hyperparameter convergence reached at iteration 2556.
The minimum predicted mean of the observed indices is -1.243 at the point
[0.383 0.    0.02  0.014 0.019 0.024]
Iteration 41
Acquisition function convergence reached at iteration 288.
The final EI loss was -0.017 with predicted mean of [-0.552]
The next parameters to simulate from are [[0.408 0.002 0.02  0.008 0.05  0.03 ]]
The mean of the samples was -0.994
Iteration 42
Acquisition function convergence reached at iteration 101.
The final EI loss was -0.022 with predicted mean of [-1.119]
The next parameters to simulate from are [[0.55  0.    0.029 0.013 0.026 0.03 ]]
The mean of the samples was -1.082
Iteration 43
```

```
Acquisition function convergence reached at iteration 1778.
The final EI loss was -0.013 with predicted mean of [0.088]
The next parameters to simulate from are [[1.    1.    0.033 0.05  0.    0.067]]
The mean of the samples was 0.761
Iteration 44
Acquisition function convergence reached at iteration 94.
The final EI loss was -0.018 with predicted mean of [-1.166]
The next parameters to simulate from are [[0.444 0.    0.016 0.014 0.017 0.024]]
The mean of the samples was -1.247
Iteration 45
Acquisition function convergence reached at iteration 114.
The final EI loss was -0.018 with predicted mean of [-0.558]
The next parameters to simulate from are [[0.497 0.    0.014 0.015 0.    0.018]]
The mean of the samples was -0.532
Iteration 46
Acquisition function convergence reached at iteration 159.
The final EI loss was -0.045 with predicted mean of [-1.225]
The next parameters to simulate from are [[0.149 0.    0.016 0.014 0.017 0.03 ]]
The mean of the samples was -1.361
Iteration 47
Acquisition function convergence reached at iteration 296.
The final EI loss was -0.035 with predicted mean of [-1.072]
The next parameters to simulate from are [[0.03  0.    0.013 0.011 0.015 0.03 ]]
The mean of the samples was -1.142
Iteration 48
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.315]
The next parameters to simulate from are [[0.418 0.    0.016 0.048 0.034 0.03 ]]
The mean of the samples was 1.527
Iteration 49
Acquisition function convergence reached at iteration 190.
The final EI loss was -0.042 with predicted mean of [-1.067]
The next parameters to simulate from are [[0.069 0.992 0.017 0.015 0.015 0.03 ]]
The mean of the samples was -1.201
Iteration 50
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.055]
The next parameters to simulate from are [[0.149 0.012 0.016 0.045 0.017 0.061]]
The mean of the samples was 1.101
Trained parameters:
amplitude_champ:0 is 0.821

length_scales_champ:0 is [0.328 2.201 0.016 0.011 0.016 0.017]
```

```
observation_noise_variance_champ:0 is 0.048

bias_mean:0 is 0.907

dictionary saved successfully to file
Iteration 51
Acquisition function convergence reached at iteration 430.
The final EI loss was -0.026 with predicted mean of [-1.079]
The next parameters to simulate from are [[0.004 0.    0.024 0.013 0.015 0.033]]
The mean of the samples was -0.981
Iteration 52
Acquisition function convergence reached at iteration 1265.
The final EI loss was -0.024 with predicted mean of [-0.897]
The next parameters to simulate from are [[0.    0.999 0.008 0.015 0.013 0.028]]
The mean of the samples was -1.216
Iteration 53
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.758]
The next parameters to simulate from are [[0.468 0.    0.016 0.037 0.017 0.03 ]]
The mean of the samples was 0.827
Iteration 54
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.696]
The next parameters to simulate from are [[0.014 0.    0.016 0.031 0.017 0.03 ]]
The mean of the samples was 0.732
Iteration 55
Acquisition function convergence reached at iteration 418.
The final EI loss was -0.018 with predicted mean of [-0.528]
The next parameters to simulate from are [[0.006 0.    0.026 0.013 0.025 0.054]]
The mean of the samples was -0.937
Iteration 56
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.819]
The next parameters to simulate from are [[0.149 0.142 0.013 0.037 0.017 0.052]]
The mean of the samples was 0.824
Iteration 57
Acquisition function convergence reached at iteration 1180.
The final EI loss was -0.029 with predicted mean of [-0.798]
The next parameters to simulate from are [[0.199 1.    0.033 0.016 0.019 0.051]]
The mean of the samples was -0.829
Iteration 58
Acquisition function convergence reached at iteration 97.
```

```
The final EI loss was -0.039 with predicted mean of [-1.202]
The next parameters to simulate from are [[0.119 0.    0.012 0.016 0.012 0.027]]
The mean of the samples was -1.297
Iteration 59
Acquisition function convergence reached at iteration 998.
The final EI loss was -0.01 with predicted mean of [-0.866]
The next parameters to simulate from are [[0.619 0.    0.033 0.015 0.024 0.019]]
The mean of the samples was -0.937
Iteration 60
Acquisition function convergence reached at iteration 1371.
The final EI loss was -0.008 with predicted mean of [0.15]
The next parameters to simulate from are [[1.    0.    0.    0.05  0.    0.067]]
The mean of the samples was 0.555
Hyperparameter convergence reached at iteration 2270.
The minimum predicted mean of the observed indices is -1.312 at the point
[0.149 0.    0.016 0.014 0.017 0.03 ]
Iteration 61
Acquisition function convergence reached at iteration 1753.
The final EI loss was -0.018 with predicted mean of [-0.54]
The next parameters to simulate from are [[0.133 0.    0.033 0.013 0.027 0.067]]
The mean of the samples was -0.809
Iteration 62
Acquisition function convergence reached at iteration 112.
The final EI loss was -0.033 with predicted mean of [-1.31]
The next parameters to simulate from are [[0.235 0.    0.014 0.015 0.015 0.026]]
The mean of the samples was -1.584
Iteration 63
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.437]
The next parameters to simulate from are [[0.235 0.    0.027 0.044 0.021 0.026]]
The mean of the samples was 1.329
Iteration 64
Acquisition function convergence reached at iteration 1009.
The final EI loss was -0.012 with predicted mean of [-0.531]
The next parameters to simulate from are [[0.    0.999 0.033 0.006 0.02  0.055]]
The mean of the samples was -0.26
Iteration 65
Acquisition function convergence reached at iteration 132.
The final EI loss was -0.014 with predicted mean of [-1.376]
The next parameters to simulate from are [[0.261 0.    0.014 0.013 0.016 0.025]]
The mean of the samples was -1.606
Iteration 66
Acquisition function convergence reached at iteration 153.
```

```
The final EI loss was -0.022 with predicted mean of [-1.243]
The next parameters to simulate from are [[0.212 1.    0.013 0.014 0.013 0.025]]
The mean of the samples was -1.05
Iteration 67
Acquisition function convergence reached at iteration 855.
The final EI loss was -0.002 with predicted mean of [0.372]
The next parameters to simulate from are [[1.    0.    0.033 0.05  0.05  0.   ]]
The mean of the samples was 1.549
Iteration 68
Acquisition function convergence reached at iteration 1177.
The final EI loss was -0.013 with predicted mean of [-0.676]
The next parameters to simulate from are [[0.    0.999 0.026 0.015 0.039 0.057]]
The mean of the samples was -0.259
Iteration 69
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.856]
The next parameters to simulate from are [[0.235 0.497 0.014 0.037 0.015 0.034]]
The mean of the samples was 0.771
Iteration 70
Acquisition function convergence reached at iteration 1073.
The final EI loss was -0.006 with predicted mean of [-0.603]
The next parameters to simulate from are [[0.    0.999 0.033 0.016 0.011 0.038]]
The mean of the samples was -0.936
Iteration 71
Acquisition function convergence reached at iteration 1386.
The final EI loss was -0.01 with predicted mean of [-0.362]
The next parameters to simulate from are [[0.251 0.    0.013 0.    0.05  0.031]]
The mean of the samples was 0.39
Iteration 72
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.193]
The next parameters to simulate from are [[0.623 0.    0.014 0.042 0.039 0.026]]
The mean of the samples was 1.386
Iteration 73
Acquisition function convergence reached at iteration 81.
The final EI loss was -0.01 with predicted mean of [-1.376]
The next parameters to simulate from are [[0.272 0.    0.014 0.016 0.015 0.025]]
The mean of the samples was -1.333
Iteration 74
Acquisition function convergence reached at iteration 622.
The final EI loss was -0.009 with predicted mean of [-0.478]
The next parameters to simulate from are [[0.28  0.    0.033 0.022 0.    0.04 ]]
The mean of the samples was -0.842
```

```
Iteration 75
Acquisition function convergence reached at iteration 999.
The final EI loss was -0.009 with predicted mean of [-0.739]
The next parameters to simulate from are [[0.    0.    0.033 0.018 0.013 0.05 ]]
The mean of the samples was -1.074
Iteration 76
Acquisition function convergence reached at iteration 791.
The final EI loss was -0.01 with predicted mean of [-1.001]
The next parameters to simulate from are [[0.527 0.001 0.023 0.01  0.042 0.027]]
The mean of the samples was -1.143
Iteration 77
Acquisition function convergence reached at iteration 1245.
The final EI loss was -0.013 with predicted mean of [-1.007]
The next parameters to simulate from are [[0.    0.    0.023 0.016 0.017 0.047]]
The mean of the samples was -0.828
Iteration 78
Acquisition function convergence reached at iteration 1617.
The final EI loss was -0.008 with predicted mean of [0.017]
The next parameters to simulate from are [[0.    0.    0.    0.    0.    0.008]]
The mean of the samples was 0.551
Iteration 79
Acquisition function convergence reached at iteration 115.
The final EI loss was -0.012 with predicted mean of [-1.088]
The next parameters to simulate from are [[0.543 0.    0.021 0.012 0.037 0.021]]
The mean of the samples was -1.24
Iteration 80
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.303]
The next parameters to simulate from are [[0.054 0.586 0.014 0.05  0.016 0.025]]
The mean of the samples was 1.65
Hyperparameter convergence reached at iteration 2938.
The minimum predicted mean of the observed indices is -1.422 at the point
[0.261 0.    0.014 0.013 0.016 0.025]
Iteration 81
Acquisition function convergence reached at iteration 1465.
The final EI loss was -0.038 with predicted mean of [-1.092]
The next parameters to simulate from are [[0.    0.997 0.015 0.016 0.007 0.03 ]]
The mean of the samples was -1.06
Iteration 82
Acquisition function convergence reached at iteration 534.
The final EI loss was -0.024 with predicted mean of [-1.13]
The next parameters to simulate from are [[0.    0.694 0.013 0.014 0.01  0.023]]
The mean of the samples was -1.025
```

```
Iteration 83
Acquisition function convergence reached at iteration 102.
The final EI loss was -0.007 with predicted mean of [-0.696]
The next parameters to simulate from are [[0.364 0.995 0.016 0.015 0.024 0.018]]
The mean of the samples was -1.094
Iteration 84
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.755]
The next parameters to simulate from are [[0.641 0.    0.018 0.037 0.016 0.025]]
The mean of the samples was 0.809
Iteration 85
Acquisition function convergence reached at iteration 1235.
The final EI loss was -0.021 with predicted mean of [-0.917]
The next parameters to simulate from are [[0.285 0.585 0.033 0.018 0.009 0.037]]
The mean of the samples was -0.958
Iteration 86
Acquisition function convergence reached at iteration 489.
The final EI loss was -0.013 with predicted mean of [-0.544]
The next parameters to simulate from are [[0.316 0.999 0.033 0.016 0.    0.025]]
The mean of the samples was -0.799
Iteration 87
Acquisition function convergence reached at iteration 1262.
The final EI loss was -0.006 with predicted mean of [-0.085]
The next parameters to simulate from are [[0.999 0.002 0.    0.019 0.    0.   ]]
The mean of the samples was 0.385
Iteration 88
Acquisition function convergence reached at iteration 94.
The final EI loss was -0.009 with predicted mean of [-0.972]
The next parameters to simulate from are [[0.479 0.    0.015 0.009 0.039 0.024]]
The mean of the samples was -1.361
Iteration 89
Acquisition function convergence reached at iteration 1077.
The final EI loss was -0.011 with predicted mean of [-0.384]
The next parameters to simulate from are [[0.521 0.998 0.025 0.016 0.    0.004]]
The mean of the samples was 0.013
Iteration 90
Acquisition function convergence reached at iteration 1271.
The final EI loss was -0.005 with predicted mean of [0.093]
The next parameters to simulate from are [[0.    0.999 0.    0.    0.013 0.067]]
The mean of the samples was 0.523
Iteration 91
Acquisition function convergence reached at iteration 108.
The final EI loss was -0.005 with predicted mean of [-0.942]
```

```
The next parameters to simulate from are [[0.475 0.001 0.015 0.017 0.011 0.02 ]]
The mean of the samples was -0.934
Iteration 92
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.959]
The next parameters to simulate from are [[0.609 0.    0.014 0.038 0.016 0.015]]
The mean of the samples was 0.929
Iteration 93
Acquisition function convergence reached at iteration 108.
The final EI loss was -0.02 with predicted mean of [-1.288]
The next parameters to simulate from are [[0.396 0.    0.02  0.009 0.04  0.024]]
The mean of the samples was -1.099
Iteration 94
Acquisition function convergence reached at iteration 1066.
The final EI loss was -0.005 with predicted mean of [0.111]
The next parameters to simulate from are [[0.999 0.    0.033 0.    0.    0.067]]
The mean of the samples was 0.603
Iteration 95
Acquisition function convergence reached at iteration 680.
The final EI loss was -0.021 with predicted mean of [-0.85]
The next parameters to simulate from are [[0.013 0.912 0.032 0.013 0.03  0.037]]
The mean of the samples was -0.81
Iteration 96
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.182]
The next parameters to simulate from are [[0.261 0.6   0.014 0.041 0.016 0.022]]
The mean of the samples was 1.165
Iteration 97
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.205]
The next parameters to simulate from are [[0.341 0.    0.014 0.044 0.016 0.025]]
The mean of the samples was 1.271
Iteration 98
Acquisition function convergence reached at iteration 1064.
The final EI loss was -0.016 with predicted mean of [-0.947]
The next parameters to simulate from are [[0.001 1.    0.025 0.013 0.009 0.028]]
The mean of the samples was -0.815
Iteration 99
Acquisition function convergence reached at iteration 89.
The final EI loss was -0.011 with predicted mean of [-1.053]
The next parameters to simulate from are [[0.091 0.999 0.015 0.017 0.014 0.022]]
The mean of the samples was -0.774
Iteration 100
```

```
Acquisition function convergence reached at iteration 742.
The final EI loss was -0.004 with predicted mean of [0.191]
The next parameters to simulate from are [[1.    0.    0.    0.009 0.    0.067]]
The mean of the samples was 0.584
Hyperparameter convergence reached at iteration 2056.
The minimum predicted mean of the observed indices is -1.425 at the point
[0.261 0.    0.014 0.013 0.016 0.025]
Trained parameters:
amplitude_champ:0 is 0.773

length_scales_champ:0 is [0.353 0.704 0.016 0.008 0.014 0.018]

observation_noise_variance_champ:0 is 0.05

bias_mean:0 is 0.852

dictionary saved successfully to file
Iteration 101
Acquisition function convergence reached at iteration 584.
The final EI loss was -0.022 with predicted mean of [-0.943]
The next parameters to simulate from are [[0.215 0.    0.033 0.018 0.018 0.055]]
The mean of the samples was -0.847
Iteration 102
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.707]
The next parameters to simulate from are [[0.261 0.66  0.014 0.031 0.024 0.025]]
The mean of the samples was 0.705
Iteration 103
Acquisition function convergence reached at iteration 90.
The final EI loss was -0.018 with predicted mean of [-1.412]
The next parameters to simulate from are [[0.207 0.    0.015 0.013 0.017 0.028]]
The mean of the samples was -1.545
Iteration 104
Acquisition function convergence reached at iteration 605.
The final EI loss was -0.003 with predicted mean of [-0.15]
The next parameters to simulate from are [[0.545 0.    0.022 0.02  0.    0.001]]
The mean of the samples was 0.434
Iteration 105
Acquisition function convergence reached at iteration 1377.
The final EI loss was -0.019 with predicted mean of [-1.013]
The next parameters to simulate from are [[0.    0.522 0.033 0.016 0.016 0.05 ]]
The mean of the samples was -1.019
Iteration 106
```

```
Acquisition function convergence reached at iteration 107.
The final EI loss was -0.004 with predicted mean of [-0.114]
The next parameters to simulate from are [[0.019 0.    0.001 0.007 0.032 0.023]]
The mean of the samples was -0.392
Iteration 107
Acquisition function convergence reached at iteration 111.
The final EI loss was -0.014 with predicted mean of [-1.413]
The next parameters to simulate from are [[0.22  0.    0.014 0.012 0.018 0.027]]
The mean of the samples was -1.301
Iteration 108
Acquisition function convergence reached at iteration 133.
The final EI loss was -0.04 with predicted mean of [-1.328]
The next parameters to simulate from are [[0.54  0.25  0.018 0.009 0.041 0.025]]
The mean of the samples was -1.052
Iteration 109
Acquisition function convergence reached at iteration 80.
The final EI loss was -0.012 with predicted mean of [-1.421]
The next parameters to simulate from are [[0.187 0.    0.015 0.013 0.014 0.027]]
The mean of the samples was -1.314
Iteration 110
Acquisition function convergence reached at iteration 120.
The final EI loss was -0.012 with predicted mean of [-1.026]
The next parameters to simulate from are [[0.495 0.    0.012 0.009 0.034 0.02 ]]
The mean of the samples was -1.675
Iteration 111
Acquisition function convergence reached at iteration 1596.
The final EI loss was -0.026 with predicted mean of [-1.038]
The next parameters to simulate from are [[0.    0.93  0.018 0.015 0.013 0.042]]
The mean of the samples was -1.124
Iteration 112
Acquisition function convergence reached at iteration 107.
The final EI loss was -0.008 with predicted mean of [-1.379]
The next parameters to simulate from are [[0.15  0.    0.014 0.014 0.017 0.028]]
The mean of the samples was -1.515
Iteration 113
Acquisition function convergence reached at iteration 940.
The final EI loss was -0.003 with predicted mean of [0.135]
The next parameters to simulate from are [[1.    0.    0.    0.    0.05  0.054]]
The mean of the samples was 0.942
Iteration 114
Acquisition function convergence reached at iteration 90.
The final EI loss was -0.008 with predicted mean of [-1.414]
The next parameters to simulate from are [[0.274 0.    0.015 0.013 0.018 0.027]]
```

```
The mean of the samples was -1.441
Iteration 115
Acquisition function convergence reached at iteration 1007.
The final EI loss was -0.018 with predicted mean of [-1.13]
The next parameters to simulate from are [[0.    0.638 0.01  0.016 0.012 0.034]]
The mean of the samples was -1.118
Iteration 116
Acquisition function convergence reached at iteration 1114.
The final EI loss was -0.005 with predicted mean of [0.06]
The next parameters to simulate from are [[1.    0.999 0.033 0.    0.    0.   ]]
The mean of the samples was 0.567
Iteration 117
Acquisition function convergence reached at iteration 831.
The final EI loss was -0.002 with predicted mean of [0.223]
The next parameters to simulate from are [[0.    0.    0.    0.027 0.05  0.   ]]
The mean of the samples was 1.051
Iteration 118
Acquisition function convergence reached at iteration 1160.
The final EI loss was -0.005 with predicted mean of [-0.281]
The next parameters to simulate from are [[0.    0.    0.    0.003 0.022 0.038]]
The mean of the samples was -0.195
Iteration 119
Acquisition function convergence reached at iteration 113.
The final EI loss was -0.019 with predicted mean of [-0.919]
The next parameters to simulate from are [[0.284 0.549 0.025 0.011 0.041 0.036]]
The mean of the samples was -0.975
Iteration 120
Acquisition function convergence reached at iteration 816.
The final EI loss was -0.002 with predicted mean of [0.199]
The next parameters to simulate from are [[0.    0.    0.033 0.025 0.    0.   ]]
The mean of the samples was 0.967
Hyperparameter convergence reached at iteration 2061.
The minimum predicted mean of the observed indices is -1.424 at the point
[0.261 0.    0.014 0.013 0.016 0.025]
Iteration 121
Acquisition function convergence reached at iteration 563.
The final EI loss was -0.008 with predicted mean of [-0.43]
The next parameters to simulate from are [[0.003 0.003 0.022 0.01  0.031 0.067]]
The mean of the samples was -0.685
Iteration 122
Acquisition function convergence reached at iteration 75.
The final EI loss was -0.01 with predicted mean of [-1.373]
The next parameters to simulate from are [[0.127 0.    0.012 0.014 0.017 0.029]]
```

```
The mean of the samples was -1.302
Iteration 123
Acquisition function convergence reached at iteration 894.
The final EI loss was -0.016 with predicted mean of [-0.618]
The next parameters to simulate from are [[0.264 1.    0.033 0.012 0.05  0.037]]
The mean of the samples was -0.785
Iteration 124
Acquisition function convergence reached at iteration 1069.
The final EI loss was -0.057 with predicted mean of [-1.471]
The next parameters to simulate from are [[0.495 0.001 0.016 0.009 0.033 0.021]]
The mean of the samples was -1.373
Iteration 125
Acquisition function convergence reached at iteration 970.
The final EI loss was -0.011 with predicted mean of [-0.59]
The next parameters to simulate from are [[0.    0.926 0.004 0.008 0.016 0.021]]
The mean of the samples was -1.354
Iteration 126
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.36]
The next parameters to simulate from are [[0.642 0.485 0.016 0.047 0.039 0.021]]
The mean of the samples was 1.326
Iteration 127
Acquisition function convergence reached at iteration 1353.
The final EI loss was -0.008 with predicted mean of [-0.396]
The next parameters to simulate from are [[0.    0.    0.033 0.019 0.033 0.067]]
The mean of the samples was -0.081
Iteration 128
Acquisition function convergence reached at iteration 949.
The final EI loss was -0.004 with predicted mean of [0.135]
The next parameters to simulate from are [[1.    0.001 0.033 0.032 0.05  0.067]]
The mean of the samples was 0.921
Iteration 129
Acquisition function convergence reached at iteration 110.
The final EI loss was -0.023 with predicted mean of [-1.345]
The next parameters to simulate from are [[0.4   0.001 0.014 0.009 0.033 0.023]]
The mean of the samples was -1.379
Iteration 130
Acquisition function convergence reached at iteration 830.
The final EI loss was -0.02 with predicted mean of [-1.229]
The next parameters to simulate from are [[0.    0.551 0.007 0.01  0.017 0.023]]
The mean of the samples was -1.485
Iteration 131
Acquisition function convergence reached at iteration 136.
```

```
The final EI loss was -0.04 with predicted mean of [-1.397]
The next parameters to simulate from are [[0.    0.615 0.007 0.007 0.021 0.022]]
The mean of the samples was -0.898
Iteration 132
Acquisition function convergence reached at iteration 889.
The final EI loss was -0.015 with predicted mean of [-1.023]
The next parameters to simulate from are [[0.179 0.999 0.024 0.017 0.012 0.036]]
The mean of the samples was -0.92
Iteration 133
Acquisition function convergence reached at iteration 142.
The final EI loss was -0.025 with predicted mean of [-1.367]
The next parameters to simulate from are [[0.432 0.    0.015 0.008 0.035 0.018]]
The mean of the samples was -1.105
Iteration 134
Acquisition function convergence reached at iteration 1122.
The final EI loss was -0.011 with predicted mean of [-0.703]
The next parameters to simulate from are [[0.001 0.385 0.033 0.009 0.039 0.044]]
The mean of the samples was -0.68
Iteration 135
Acquisition function convergence reached at iteration 940.
The final EI loss was -0.036 with predicted mean of [-1.466]
The next parameters to simulate from are [[0.493 0.    0.014 0.009 0.032 0.026]]
The mean of the samples was -1.47
Iteration 136
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.152]
The next parameters to simulate from are [[0.495 0.488 0.016 0.043 0.022 0.01 ]]
The mean of the samples was 1.232
Iteration 137
Acquisition function convergence reached at iteration 176.
The final EI loss was -0.027 with predicted mean of [-1.477]
The next parameters to simulate from are [[0.494 0.    0.015 0.01  0.031 0.023]]
The mean of the samples was -1.537
Iteration 138
Acquisition function convergence reached at iteration 113.
The final EI loss was -0.014 with predicted mean of [-1.438]
The next parameters to simulate from are [[0.44  0.    0.014 0.01  0.029 0.024]]
The mean of the samples was -1.544
Iteration 139
Acquisition function convergence reached at iteration 1254.
The final EI loss was -0.015 with predicted mean of [-0.769]
The next parameters to simulate from are [[0.    0.961 0.019 0.007 0.038 0.037]]
The mean of the samples was -0.88
```

```
Iteration 140
Acquisition function convergence reached at iteration 657.
The final EI loss was -0.001 with predicted mean of [0.357]
The next parameters to simulate from are [[0.999 0.    0.    0.05  0.    0.036]]
The mean of the samples was 0.68
Hyperparameter convergence reached at iteration 2074.
The minimum predicted mean of the observed indices is -1.521 at the point
[0.4   0.001 0.014 0.01  0.033 0.023]
Iteration 141
Acquisition function convergence reached at iteration 111.
The final EI loss was -0.02 with predicted mean of [-1.474]
The next parameters to simulate from are [[0.412 0.    0.013 0.01  0.035 0.026]]
The mean of the samples was -1.273
Iteration 142
Acquisition function convergence reached at iteration 1017.
The final EI loss was -0.005 with predicted mean of [0.067]
The next parameters to simulate from are [[0.998 0.001 0.033 0.029 0.    0.067]]
The mean of the samples was 0.427
Iteration 143
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.787]
The next parameters to simulate from are [[0.44  0.    0.014 0.03  0.029 0.024]]
The mean of the samples was 0.8
Iteration 144
Acquisition function convergence reached at iteration 135.
The final EI loss was -0.024 with predicted mean of [-1.305]
The next parameters to simulate from are [[0.147 0.    0.009 0.01  0.022 0.025]]
The mean of the samples was -1.594
Iteration 145
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.52]
The next parameters to simulate from are [[0.439 0.09  0.019 0.01  0.01  0.024]]
The mean of the samples was -0.542
Iteration 146
Acquisition function convergence reached at iteration 128.
The final EI loss was -0.026 with predicted mean of [-1.499]
The next parameters to simulate from are [[0.222 0.    0.01  0.01  0.025 0.025]]
The mean of the samples was -1.741
Iteration 147
Acquisition function convergence reached at iteration 1393.
The final EI loss was -0.02 with predicted mean of [-1.386]
The next parameters to simulate from are [[0.    0.998 0.007 0.011 0.018 0.025]]
The mean of the samples was -1.672
```

```
Iteration 148
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.041]
The next parameters to simulate from are [[0.824 0.184 0.007 0.011 0.018 0.025]]
The mean of the samples was 0.036
Iteration 149
Acquisition function convergence reached at iteration 123.
The final EI loss was -0.025 with predicted mean of [-1.531]
The next parameters to simulate from are [[0.11  0.758 0.008 0.011 0.02  0.024]]
The mean of the samples was -1.55
Iteration 150
Acquisition function convergence reached at iteration 756.
The final EI loss was -0.002 with predicted mean of [0.151]
The next parameters to simulate from are [[0.    0.    0.021 0.    0.    0.067]]
The mean of the samples was 0.61
Trained parameters:
amplitude_champ:0 is 0.805

length_scales_champ:0 is [0.378 1.695 0.012 0.006 0.017 0.019]

observation_noise_variance_champ:0 is 0.062

bias_mean:0 is 0.783

dictionary saved successfully to file
Iteration 151
Acquisition function convergence reached at iteration 657.
The final EI loss was -0.001 with predicted mean of [0.377]
The next parameters to simulate from are [[0.   1.   0.   0.05 0.05 0.  ]]
The mean of the samples was 2.075
Iteration 152
Acquisition function convergence reached at iteration 466.
The final EI loss was -0.003 with predicted mean of [-0.368]
The next parameters to simulate from are [[0.365 0.    0.033 0.008 0.031 0.067]]
The mean of the samples was -0.396
Iteration 153
Acquisition function convergence reached at iteration 564.
The final EI loss was -0.001 with predicted mean of [0.4]
The next parameters to simulate from are [[0.   1.   0.   0.05 0.   0.  ]]
The mean of the samples was 2.094
Iteration 154
Acquisition function convergence reached at iteration 67.
The final EI loss was -0.005 with predicted mean of [-0.532]
```

```
The next parameters to simulate from are [[0.    0.99  0.025 0.019 0.002 0.052]]
The mean of the samples was -0.716
Iteration 155
Acquisition function convergence reached at iteration 592.
The final EI loss was -0.001 with predicted mean of [0.061]
The next parameters to simulate from are [[0.    1.    0.    0.    0.026 0.017]]
The mean of the samples was 0.459
Iteration 156
Acquisition function convergence reached at iteration 1085.
The final EI loss was -0.004 with predicted mean of [-0.547]
The next parameters to simulate from are [[0.    1.    0.033 0.022 0.    0.034]]
The mean of the samples was -0.704
Iteration 157
Acquisition function convergence reached at iteration 556.
The final EI loss was -0.001 with predicted mean of [0.195]
The next parameters to simulate from are [[0.    0.001 0.    0.025 0.    0.045]]
The mean of the samples was 0.409
Iteration 158
Acquisition function convergence reached at iteration 133.
The final EI loss was -0.018 with predicted mean of [-1.525]
The next parameters to simulate from are [[0.    1.    0.006 0.011 0.015 0.023]]
The mean of the samples was -1.567
Iteration 159
Acquisition function convergence reached at iteration 688.
The final EI loss was -0.002 with predicted mean of [-0.108]
The next parameters to simulate from are [[1.    0.999 0.011 0.013 0.    0.   ]]
The mean of the samples was 0.29
Iteration 160
Acquisition function convergence reached at iteration 494.
The final EI loss was -0.001 with predicted mean of [0.407]
The next parameters to simulate from are [[1.    0.998 0.033 0.037 0.05  0.035]]
The mean of the samples was 0.666
Hyperparameter convergence reached at iteration 1929.
The minimum predicted mean of the observed indices is -1.617 at the point
[0.    0.998 0.007 0.011 0.018 0.025]
Iteration 161
Acquisition function convergence reached at iteration 82.
The final EI loss was -0.018 with predicted mean of [-1.591]
The next parameters to simulate from are [[0.    0.584 0.007 0.011 0.016 0.025]]
The mean of the samples was -1.795
Iteration 162
Acquisition function convergence reached at iteration 125.
The final EI loss was -0.001 with predicted mean of [-0.265]
```

```
The next parameters to simulate from are [[0.626 0.995 0.011 0.019 0.022 0.007]]
The mean of the samples was -0.849
Iteration 163
Acquisition function convergence reached at iteration 683.
The final EI loss was -0.001 with predicted mean of [0.274]
The next parameters to simulate from are [[0.999 0.999 0.033 0.    0.    0.033]]
The mean of the samples was 0.569
Iteration 164
Acquisition function convergence reached at iteration 410.
The final EI loss was -0.001 with predicted mean of [-0.091]
The next parameters to simulate from are [[0.631 0.998 0.    0.019 0.025 0.   ]]
The mean of the samples was 0.384
Iteration 165
Acquisition function convergence reached at iteration 136.
The final EI loss was -0.004 with predicted mean of [-0.607]
The next parameters to simulate from are [[0.002 0.992 0.021 0.008 0.05  0.048]]
The mean of the samples was -0.777
Iteration 166
Acquisition function convergence reached at iteration 494.
The final EI loss was -0.001 with predicted mean of [0.278]
The next parameters to simulate from are [[0.    0.998 0.    0.034 0.    0.067]]
The mean of the samples was 0.505
Iteration 167
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.682]
The next parameters to simulate from are [[0.715 0.584 0.008 0.037 0.016 0.025]]
The mean of the samples was 0.579
Iteration 168
Acquisition function convergence reached at iteration 569.
The final EI loss was -0.001 with predicted mean of [0.372]
The next parameters to simulate from are [[0.    0.999 0.    0.05  0.05  0.067]]
The mean of the samples was 1.939
Iteration 169
Acquisition function convergence reached at iteration 132.
The final EI loss was -0.018 with predicted mean of [-1.639]
The next parameters to simulate from are [[0.    0.58  0.007 0.011 0.016 0.026]]
The mean of the samples was -1.668
Iteration 170
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.468]
The next parameters to simulate from are [[0.    0.584 0.015 0.026 0.016 0.017]]
The mean of the samples was 0.51
Iteration 171
```

```
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.054]
The next parameters to simulate from are [[0.299 0.584 0.019 0.004 0.016 0.025]]
The mean of the samples was -0.017
Iteration 172
Acquisition function convergence reached at iteration 51.
The final EI loss was -0.002 with predicted mean of [-0.746]
The next parameters to simulate from are [[0.257 0.924 0.014 0.009 0.047 0.036]]
The mean of the samples was -0.827
Iteration 173
Acquisition function convergence reached at iteration 739.
The final EI loss was -0.001 with predicted mean of [0.119]
The next parameters to simulate from are [[0.    0.    0.033 0.    0.    0.   ]]
The mean of the samples was 0.526
Iteration 174
Acquisition function convergence reached at iteration 97.
The final EI loss was -0.002 with predicted mean of [-0.781]
The next parameters to simulate from are [[0.    0.941 0.019 0.008 0.029 0.051]]
The mean of the samples was -0.768
Iteration 175
Acquisition function convergence reached at iteration 154.
The final EI loss was -0.009 with predicted mean of [-1.666]
The next parameters to simulate from are [[0.    0.572 0.007 0.011 0.013 0.026]]
The mean of the samples was -1.476
Iteration 176
Acquisition function convergence reached at iteration 803.
The final EI loss was -0.002 with predicted mean of [0.083]
The next parameters to simulate from are [[1.    0.001 0.033 0.    0.05  0.   ]]
The mean of the samples was 0.546
Iteration 177
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.064]
The next parameters to simulate from are [[0.651 0.585 0.007 0.05  0.016 0.006]]
The mean of the samples was 1.315
Iteration 178
Acquisition function convergence reached at iteration 472.
The final EI loss was -0.012 with predicted mean of [-1.681]
The next parameters to simulate from are [[0.    0.513 0.006 0.011 0.018 0.024]]
The mean of the samples was -1.782
Iteration 179
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.321]
The next parameters to simulate from are [[0.735 0.417 0.007 0.011 0.016 0.048]]
```

```
The mean of the samples was 0.288
Iteration 180
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.018]
The next parameters to simulate from are [[0.    0.584 0.022 0.033 0.032 0.025]]
The mean of the samples was 1.078
Hyperparameter convergence reached at iteration 2188.
The minimum predicted mean of the observed indices is -1.736 at the point
[0.    0.584 0.007 0.011 0.016 0.025]
Iteration 181
Acquisition function convergence reached at iteration 624.
The final EI loss was -0.004 with predicted mean of [-1.157]
The next parameters to simulate from are [[0.    0.001 0.011 0.009 0.028 0.033]]
The mean of the samples was -1.392
Iteration 182
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.197]
The next parameters to simulate from are [[0.68  0.151 0.007 0.011 0.016 0.044]]
The mean of the samples was 0.088
Iteration 183
Acquisition function convergence reached at iteration 238.
The final EI loss was -0.0 with predicted mean of [0.445]
The next parameters to simulate from are [[0.    0.005 0.    0.034 0.    0.037]]
The mean of the samples was 0.651
Iteration 184
Acquisition function convergence reached at iteration 99.
The final EI loss was -0.008 with predicted mean of [-1.689]
The next parameters to simulate from are [[0.    0.271 0.007 0.011 0.018 0.027]]
The mean of the samples was -1.753
Iteration 185
Acquisition function convergence reached at iteration 128.
The final EI loss was -0.008 with predicted mean of [-1.642]
The next parameters to simulate from are [[0.079 0.333 0.007 0.011 0.017 0.023]]
The mean of the samples was -1.862
Iteration 186
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.349]
The next parameters to simulate from are [[0.    0.584 0.007 0.011 0.045 0.025]]
The mean of the samples was -0.274
Iteration 187
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.897]
The next parameters to simulate from are [[0.96  0.457 0.009 0.047 0.016 0.006]]
```

```
The mean of the samples was 0.949
Iteration 188
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.127]
The next parameters to simulate from are [[0.    0.299 0.007 0.011 0.    0.025]]
The mean of the samples was -0.137
Iteration 189
WARNING:tensorflow:5 out of the last 135 calls to <function update_var_EI.<locals>.opt_var at
Acquisition function convergence reached at iteration 442.
The final EI loss was -0.001 with predicted mean of [0.252]
The next parameters to simulate from are [[0.999 0.998 0.    0.025 0.014 0.067]]
The mean of the samples was 0.507
Iteration 190
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.574]
The next parameters to simulate from are [[0.101 0.413 0.007 0.046 0.014 0.025]]
The mean of the samples was 1.504
Iteration 191
Acquisition function convergence reached at iteration 462.
The final EI loss was -0.001 with predicted mean of [-0.194]
The next parameters to simulate from are [[0.001 0.999 0.017 0.021 0.007 0.067]]
The mean of the samples was -0.408
Iteration 192
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.698]
The next parameters to simulate from are [[0.522 0.584 0.015 0.011 0.016 0.025]]
The mean of the samples was -0.495
Iteration 193
Acquisition function convergence reached at iteration 359.
The final EI loss was -0.002 with predicted mean of [-1.03]
The next parameters to simulate from are [[0.652 0.001 0.012 0.015 0.015 0.01 ]]
The mean of the samples was -1.362
Iteration 194
Acquisition function convergence reached at iteration 529.
The final EI loss was -0.002 with predicted mean of [-1.079]
The next parameters to simulate from are [[0.    0.001 0.013 0.01  0.027 0.044]]
The mean of the samples was -1.134
Iteration 195
Acquisition function convergence reached at iteration 382.
The final EI loss was -0.005 with predicted mean of [-1.65]
The next parameters to simulate from are [[0.085 0.003 0.007 0.011 0.017 0.025]]
The mean of the samples was -1.695
Iteration 196
```

```
Acquisition function convergence reached at iteration 465.
The final EI loss was -0.002 with predicted mean of [-0.888]
The next parameters to simulate from are [[0.523 1.    0.026 0.01  0.05  0.03 ]]
The mean of the samples was -0.514
Iteration 197
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.139]
The next parameters to simulate from are [[0.    0.584 0.007 0.022 0.016 0.025]]
The mean of the samples was 0.074
Iteration 198
Acquisition function convergence reached at iteration 104.
The final EI loss was -0.007 with predicted mean of [-1.724]
The next parameters to simulate from are [[0.    0.313 0.006 0.011 0.017 0.024]]
The mean of the samples was -1.737
Iteration 199
Acquisition function convergence reached at iteration 498.
The final EI loss was -0.001 with predicted mean of [0.234]
The next parameters to simulate from are [[0.663 0.998 0.    0.015 0.05  0.067]]
The mean of the samples was 0.596
Iteration 200
Acquisition function convergence reached at iteration 301.
The final EI loss was -0.001 with predicted mean of [-0.863]
The next parameters to simulate from are [[0.784 0.001 0.011 0.014 0.014 0.001]]
The mean of the samples was 0.04
Hyperparameter convergence reached at iteration 2026.
The minimum predicted mean of the observed indices is -1.762 at the point
[0.    0.584 0.007 0.011 0.016 0.025]
Trained parameters:
amplitude_champ:0 is 0.796

length_scales_champ:0 is [0.364 1.127 0.01  0.006 0.015 0.016]

observation_noise_variance_champ:0 is 0.063

bias_mean:0 is 0.859

dictionary saved successfully to file
Iteration 201
Acquisition function convergence reached at iteration 490.
The final EI loss was -0.001 with predicted mean of [0.266]
The next parameters to simulate from are [[0.    0.998 0.019 0.    0.    0.   ]]
The mean of the samples was 0.524
Iteration 202
```

```
Acquisition function convergence reached at iteration 106.
The final EI loss was -0.011 with predicted mean of [-1.676]
The next parameters to simulate from are [[0.    0.656 0.004 0.011 0.017 0.023]]
The mean of the samples was -1.619
Iteration 203
Acquisition function convergence reached at iteration 635.
The final EI loss was -0.002 with predicted mean of [-0.899]
The next parameters to simulate from are [[0.39  0.001 0.011 0.009 0.05  0.028]]
The mean of the samples was -0.735
Iteration 204
Acquisition function convergence reached at iteration 568.
The final EI loss was -0.001 with predicted mean of [0.147]
The next parameters to simulate from are [[1.    0.999 0.033 0.027 0.02  0.   ]]
The mean of the samples was 0.43
Iteration 205
Acquisition function convergence reached at iteration 797.
The final EI loss was -0.001 with predicted mean of [-0.071]
The next parameters to simulate from are [[0.999 0.999 0.033 0.013 0.05  0.034]]
The mean of the samples was 0.308
Iteration 206
Acquisition function convergence reached at iteration 95.
The final EI loss was -0.009 with predicted mean of [-1.736]
The next parameters to simulate from are [[0.    0.393 0.008 0.012 0.018 0.025]]
The mean of the samples was -1.529
Iteration 207
Acquisition function convergence reached at iteration 132.
The final EI loss was -0.004 with predicted mean of [-1.337]
The next parameters to simulate from are [[0.313 0.428 0.01  0.011 0.027 0.022]]
The mean of the samples was -1.372
Iteration 208
Acquisition function convergence reached at iteration 83.
The final EI loss was -0.007 with predicted mean of [-1.719]
The next parameters to simulate from are [[0.    0.662 0.007 0.011 0.016 0.022]]
The mean of the samples was -1.59
Iteration 209
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.537]
The next parameters to simulate from are [[0.    0.584 0.007 0.027 0.016 0.025]]
The mean of the samples was 0.549
Iteration 210
Acquisition function convergence reached at iteration 546.
The final EI loss was -0.001 with predicted mean of [0.141]
The next parameters to simulate from are [[0.    0.002 0.    0.005 0.05  0.067]]
```

```
The mean of the samples was 0.569
Iteration 211
Acquisition function convergence reached at iteration 312.
The final EI loss was -0.002 with predicted mean of [-0.983]
The next parameters to simulate from are [[0.    0.999 0.023 0.013 0.022 0.037]]
The mean of the samples was -1.091
Iteration 212
Acquisition function convergence reached at iteration 164.
The final EI loss was -0.01 with predicted mean of [-1.695]
The next parameters to simulate from are [[0.082 0.673 0.006 0.011 0.016 0.024]]
The mean of the samples was -1.701
Iteration 213
Acquisition function convergence reached at iteration 295.
The final EI loss was -0.005 with predicted mean of [-1.55]
The next parameters to simulate from are [[0.    0.999 0.008 0.011 0.015 0.029]]
The mean of the samples was -1.497
Iteration 214
Acquisition function convergence reached at iteration 856.
The final EI loss was -0.005 with predicted mean of [-0.853]
The next parameters to simulate from are [[0.446 1.    0.033 0.015 0.021 0.031]]
The mean of the samples was -0.923
Iteration 215
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.351]
The next parameters to simulate from are [[0.844 0.413 0.007 0.011 0.016 0.048]]
The mean of the samples was 0.356
Iteration 216
Acquisition function convergence reached at iteration 236.
The final EI loss was -0.002 with predicted mean of [-0.613]
The next parameters to simulate from are [[0.    0.001 0.024 0.008 0.05  0.046]]
The mean of the samples was -0.698
Iteration 217
Acquisition function convergence reached at iteration 848.
The final EI loss was -0.006 with predicted mean of [-1.588]
The next parameters to simulate from are [[0.    0.001 0.007 0.011 0.017 0.023]]
The mean of the samples was -1.599
Iteration 218
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.588]
The next parameters to simulate from are [[0.034 0.98  0.006 0.029 0.013 0.025]]
The mean of the samples was 0.579
Iteration 219
Acquisition function convergence reached at iteration 462.
```

```
The final EI loss was -0.001 with predicted mean of [0.11]
The next parameters to simulate from are [[0.    0.001 0.033 0.011 0.    0.004]]
The mean of the samples was 0.232
Iteration 220
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.798]
The next parameters to simulate from are [[0.    0.761 0.018 0.011 0.016 0.025]]
The mean of the samples was -0.825
Hyperparameter convergence reached at iteration 2112.
The minimum predicted mean of the observed indices is -1.737 at the point
[0.    0.584 0.007 0.011 0.016 0.025]
Iteration 221
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.867]
The next parameters to simulate from are [[0.089 0.868 0.028 0.046 0.016 0.066]]
The mean of the samples was 1.049
Iteration 222
Acquisition function convergence reached at iteration 472.
The final EI loss was -0.001 with predicted mean of [0.208]
The next parameters to simulate from are [[1.    0.998 0.019 0.036 0.    0.067]]
The mean of the samples was 0.399
Iteration 223
Acquisition function convergence reached at iteration 636.
The final EI loss was -0.002 with predicted mean of [-0.202]
The next parameters to simulate from are [[0.    0.    0.033 0.003 0.05  0.033]]
The mean of the samples was -0.112
Iteration 224
Acquisition function convergence reached at iteration 330.
The final EI loss was -0.0 with predicted mean of [0.274]
The next parameters to simulate from are [[0.414 0.998 0.    0.014 0.    0.067]]
The mean of the samples was 0.527
Iteration 225
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.439]
The next parameters to simulate from are [[0.    0.868 0.007 0.049 0.023 0.008]]
The mean of the samples was 1.899
Iteration 226
Acquisition function convergence reached at iteration 212.
The final EI loss was -0.008 with predicted mean of [-1.704]
The next parameters to simulate from are [[0.092 0.412 0.008 0.011 0.018 0.025]]
The mean of the samples was -1.612
Iteration 227
Acquisition function convergence reached at iteration 2.
```

```
The final EI loss was -0.0 with predicted mean of [-0.36]
The next parameters to simulate from are [[0.    0.548 0.004 0.011 0.016 0.052]]
The mean of the samples was -0.427
Iteration 228
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.489]
The next parameters to simulate from are [[0.196 0.993 0.015 0.046 0.031 0.025]]
The mean of the samples was 1.464
Iteration 229
Acquisition function convergence reached at iteration 605.
The final EI loss was -0.006 with predicted mean of [-1.544]
The next parameters to simulate from are [[0.094 0.998 0.006 0.011 0.017 0.023]]
The mean of the samples was -1.62
Iteration 230
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.146]
The next parameters to simulate from are [[0.874 0.039 0.007 0.011 0.02  0.032]]
The mean of the samples was 0.109
Iteration 231
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.431]
The next parameters to simulate from are [[0.544 0.584 0.02  0.01  0.016 0.025]]
The mean of the samples was -0.429
Iteration 232
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.292]
The next parameters to simulate from are [[0.179 0.129 0.007 0.019 0.016 0.025]]
The mean of the samples was -0.259
Iteration 233
Acquisition function convergence reached at iteration 270.
The final EI loss was -0.003 with predicted mean of [-0.809]
The next parameters to simulate from are [[0.    0.001 0.029 0.018 0.    0.039]]
The mean of the samples was -0.884
Iteration 234
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.    0.584 0.007 0.011 0.007 0.025]]
The mean of the samples was -0.739
Iteration 235
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.313]
The next parameters to simulate from are [[0.224 0.409 0.015 0.041 0.016 0.015]]
The mean of the samples was 1.276
```

```
Iteration 236
Acquisition function convergence reached at iteration 329.
The final EI loss was -0.001 with predicted mean of [0.015]
The next parameters to simulate from are [[0.405 0.002 0.024 0.018 0.05  0.067]]
The mean of the samples was 0.349
Iteration 237
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.15]
The next parameters to simulate from are [[0.781 0.886 0.007 0.011 0.044 0.025]]
The mean of the samples was 0.229
Iteration 238
Acquisition function convergence reached at iteration 472.
The final EI loss was -0.002 with predicted mean of [-0.756]
The next parameters to simulate from are [[0.531 0.004 0.033 0.019 0.01  0.028]]
The mean of the samples was -0.75
Iteration 239
Acquisition function convergence reached at iteration 664.
The final EI loss was -0.002 with predicted mean of [-0.691]
The next parameters to simulate from are [[0.628 0.998 0.033 0.013 0.031 0.019]]
The mean of the samples was -0.793
Iteration 240
Acquisition function convergence reached at iteration 129.
The final EI loss was -0.008 with predicted mean of [-1.354]
The next parameters to simulate from are [[0.535 0.265 0.01  0.014 0.02  0.013]]
The mean of the samples was -1.465
Hyperparameter convergence reached at iteration 2073.
The minimum predicted mean of the observed indices is -1.735 at the point
[0.    0.584 0.007 0.011 0.016 0.025]
Iteration 241
Acquisition function convergence reached at iteration 465.
The final EI loss was -0.003 with predicted mean of [-1.641]
The next parameters to simulate from are [[0.    0.999 0.005 0.011 0.017 0.024]]
The mean of the samples was -1.911
Iteration 242
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.462]
The next parameters to simulate from are [[0.915 0.821 0.008 0.011 0.012 0.034]]
The mean of the samples was 0.376
Iteration 243
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.471]
The next parameters to simulate from are [[0.    0.816 0.016 0.045 0.008 0.008]]
The mean of the samples was 1.703
```

```
Iteration 244
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.576]
The next parameters to simulate from are [[0.    0.726 0.007 0.049 0.016 0.026]]
The mean of the samples was 1.637
Iteration 245
Acquisition function convergence reached at iteration 495.
The final EI loss was -0.001 with predicted mean of [-0.097]
The next parameters to simulate from are [[0.001 0.001 0.019 0.002 0.029 0.067]]
The mean of the samples was 0.022
Iteration 246
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.469]
The next parameters to simulate from are [[0.    0.81  0.006 0.011 0.038 0.028]]
The mean of the samples was -0.335
Iteration 247
Acquisition function convergence reached at iteration 568.
The final EI loss was -0.001 with predicted mean of [-0.826]
The next parameters to simulate from are [[0.566 0.002 0.029 0.009 0.037 0.02 ]]
The mean of the samples was -0.9
Iteration 248
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.009]
The next parameters to simulate from are [[0.    0.584 0.007 0.021 0.016 0.033]]
The mean of the samples was -0.012
Iteration 249
Acquisition function convergence reached at iteration 242.
The final EI loss was -0.001 with predicted mean of [-0.538]
The next parameters to simulate from are [[0.637 0.002 0.033 0.009 0.023 0.018]]
The mean of the samples was -0.68
Iteration 250
Acquisition function convergence reached at iteration 486.
The final EI loss was -0.002 with predicted mean of [-1.142]
The next parameters to simulate from are [[0.    0.998 0.014 0.012 0.024 0.037]]
The mean of the samples was -1.439
Trained parameters:
amplitude_champ:0 is 0.762

length_scales_champ:0 is [0.362 1.238 0.009 0.006 0.015 0.016]

observation_noise_variance_champ:0 is 0.07

bias_mean:0 is 0.867
```

```
dictionary saved successfully to file
Iteration 251
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.352]
The next parameters to simulate from are [[0.996 0.584 0.013 0.011 0.016 0.025]]
The mean of the samples was 0.383
Iteration 252
Acquisition function convergence reached at iteration 429.
The final EI loss was -0.0 with predicted mean of [0.257]
The next parameters to simulate from are [[1.    0.998 0.    0.035 0.    0.038]]
The mean of the samples was 0.471
Iteration 253
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.301]
The next parameters to simulate from are [[0.888 0.584 0.002 0.013 0.016 0.025]]
The mean of the samples was 0.338
Iteration 254
Acquisition function convergence reached at iteration 64.
The final EI loss was -0.001 with predicted mean of [-0.927]
The next parameters to simulate from are [[0.214 0.983 0.029 0.013 0.02  0.041]]
The mean of the samples was -0.894
Iteration 255
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.541]
The next parameters to simulate from are [[0.686 0.584 0.007 0.037 0.016 0.034]]
The mean of the samples was 0.525
Iteration 256
Acquisition function convergence reached at iteration 81.
The final EI loss was -0.005 with predicted mean of [-1.73]
The next parameters to simulate from are [[0.    0.41  0.006 0.01  0.017 0.025]]
The mean of the samples was -1.764
Iteration 257
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.567]
The next parameters to simulate from are [[0.    0.231 0.007 0.028 0.016 0.025]]
The mean of the samples was 0.607
Iteration 258
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.288]
The next parameters to simulate from are [[0.869 0.584 0.015 0.011 0.024 0.041]]
The mean of the samples was 0.275
Iteration 259
```

```
Acquisition function convergence reached at iteration 577.
The final EI loss was -0.002 with predicted mean of [-0.914]
The next parameters to simulate from are [[0.    0.997 0.027 0.014 0.01  0.048]]
The mean of the samples was -0.86
Iteration 260
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.066]
The next parameters to simulate from are [[0.    0.06  0.007 0.022 0.016 0.025]]
The mean of the samples was 0.084
Hyperparameter convergence reached at iteration 2135.
The minimum predicted mean of the observed indices is -1.74 at the point
[0.    0.584 0.007 0.011 0.016 0.025]
Iteration 261
Acquisition function convergence reached at iteration 415.
The final EI loss was -0.0 with predicted mean of [0.257]
The next parameters to simulate from are [[0.998 0.002 0.    0.019 0.    0.039]]
The mean of the samples was 0.448
Iteration 262
Acquisition function convergence reached at iteration 92.
The final EI loss was -0.006 with predicted mean of [-1.749]
The next parameters to simulate from are [[0.042 0.429 0.006 0.01  0.018 0.025]]
The mean of the samples was -1.688
Iteration 263
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.725]
The next parameters to simulate from are [[0.86  0.429 0.006 0.042 0.018 0.025]]
The mean of the samples was 0.788
Iteration 264
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.336]
The next parameters to simulate from are [[0.042 0.275 0.006 0.01  0.018 0.061]]
The mean of the samples was -0.36
Iteration 265
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.413]
The next parameters to simulate from are [[0.042 0.112 0.006 0.01  0.042 0.025]]
The mean of the samples was -0.479
Iteration 266
Acquisition function convergence reached at iteration 467.
The final EI loss was -0.003 with predicted mean of [-1.384]
The next parameters to simulate from are [[0.242 0.001 0.01  0.009 0.029 0.029]]
The mean of the samples was -1.469
Iteration 267
```

```
Acquisition function convergence reached at iteration 274.
The final EI loss was -0.002 with predicted mean of [-1.23]
The next parameters to simulate from are [[0.208 0.998 0.009 0.01  0.026 0.024]]
The mean of the samples was -1.227
Iteration 268
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.257]
The next parameters to simulate from are [[0.146 0.429 0.006 0.01  0.018 0.06 ]]
The mean of the samples was -0.239
Iteration 269
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.5]
The next parameters to simulate from are [[0.042 0.43  0.005 0.01  0.034 0.036]]
The mean of the samples was -0.486
Iteration 270
Acquisition function convergence reached at iteration 347.
The final EI loss was -0.002 with predicted mean of [-1.59]
The next parameters to simulate from are [[0.    0.999 0.005 0.011 0.016 0.022]]
The mean of the samples was -1.681
Iteration 271
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.663]
The next parameters to simulate from are [[0.042 0.429 0.006 0.01  0.018 0.044]]
The mean of the samples was -0.724
Iteration 272
Acquisition function convergence reached at iteration 750.
The final EI loss was -0.002 with predicted mean of [-0.426]
The next parameters to simulate from are [[0.    0.998 0.033 0.018 0.007 0.067]]
The mean of the samples was -0.824
Iteration 273
Acquisition function convergence reached at iteration 90.
The final EI loss was -0.004 with predicted mean of [-1.744]
The next parameters to simulate from are [[0.044 0.419 0.006 0.01  0.018 0.025]]
The mean of the samples was -1.907
Iteration 274
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.396]
The next parameters to simulate from are [[0.711 0.419 0.001 0.01  0.018 0.059]]
The mean of the samples was 0.408
Iteration 275
Acquisition function convergence reached at iteration 88.
The final EI loss was -0.003 with predicted mean of [-1.741]
The next parameters to simulate from are [[0.049 0.354 0.006 0.01  0.018 0.026]]
```

```
The mean of the samples was -1.816
Iteration 276
Acquisition function convergence reached at iteration 399.
The final EI loss was -0.001 with predicted mean of [-0.547]
The next parameters to simulate from are [[0.663 0.998 0.033 0.016 0.013 0.02 ]]
The mean of the samples was -0.739
Iteration 277
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.176]
The next parameters to simulate from are [[0.702 0.207 0.006 0.01  0.018 0.025]]
The mean of the samples was -0.187
Iteration 278
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.302]
The next parameters to simulate from are [[0.894 0.387 0.006 0.01  0.011 0.025]]
The mean of the samples was 0.296
Iteration 279
Acquisition function convergence reached at iteration 85.
The final EI loss was -0.003 with predicted mean of [-1.755]
The next parameters to simulate from are [[0.049 0.42  0.006 0.01  0.018 0.024]]
The mean of the samples was -1.93
Iteration 280
Acquisition function convergence reached at iteration 89.
The final EI loss was -0.002 with predicted mean of [-1.276]
The next parameters to simulate from are [[0.56  0.149 0.009 0.017 0.015 0.013]]
The mean of the samples was -0.966
Hyperparameter convergence reached at iteration 2093.
The minimum predicted mean of the observed indices is -1.777 at the point
[0.044 0.419 0.006 0.01  0.018 0.025]
Iteration 281
Acquisition function convergence reached at iteration 473.
The final EI loss was -0.001 with predicted mean of [-0.01]
The next parameters to simulate from are [[0.    0.998 0.014 0.01  0.05  0.   ]]
The mean of the samples was 0.389
Iteration 282
Acquisition function convergence reached at iteration 419.
The final EI loss was -0.0 with predicted mean of [0.196]
The next parameters to simulate from are [[0.999 0.001 0.033 0.014 0.    0.067]]
The mean of the samples was 0.466
Iteration 283
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.746]
The next parameters to simulate from are [[0.044 0.419 0.006 0.01  0.034 0.025]]
```

```
The mean of the samples was -0.759
Iteration 284
Acquisition function convergence reached at iteration 611.
The final EI loss was -0.002 with predicted mean of [-0.694]
The next parameters to simulate from are [[0.    0.999 0.027 0.014 0.015 0.067]]
The mean of the samples was -0.705
Iteration 285
Acquisition function convergence reached at iteration 491.
The final EI loss was -0.001 with predicted mean of [-0.83]
The next parameters to simulate from are [[0.    0.998 0.023 0.018 0.    0.036]]
The mean of the samples was -0.87
Iteration 286
Acquisition function convergence reached at iteration 507.
The final EI loss was -0.001 with predicted mean of [0.143]
The next parameters to simulate from are [[0.999 0.002 0.023 0.    0.05  0.067]]
The mean of the samples was 0.469
Iteration 287
Acquisition function convergence reached at iteration 90.
The final EI loss was -0.003 with predicted mean of [-1.751]
The next parameters to simulate from are [[0.061 0.261 0.006 0.01  0.018 0.025]]
The mean of the samples was -1.721
Iteration 288
Acquisition function convergence reached at iteration 86.
The final EI loss was -0.003 with predicted mean of [-1.738]
The next parameters to simulate from are [[0.055 0.325 0.005 0.01  0.018 0.023]]
The mean of the samples was -1.826
Iteration 289
Acquisition function convergence reached at iteration 336.
The final EI loss was -0.0 with predicted mean of [0.047]
The next parameters to simulate from are [[0.574 0.997 0.    0.007 0.023 0.   ]]
The mean of the samples was 0.383
Iteration 290
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.847]
The next parameters to simulate from are [[0.044 0.838 0.022 0.026 0.018 0.003]]
The mean of the samples was 0.907
Iteration 291
Acquisition function convergence reached at iteration 352.
The final EI loss was -0.003 with predicted mean of [-1.301]
The next parameters to simulate from are [[0.522 0.002 0.01  0.012 0.015 0.013]]
The mean of the samples was -1.569
Iteration 292
Acquisition function convergence reached at iteration 773.
```

```
The final EI loss was -0.002 with predicted mean of [-0.419]
The next parameters to simulate from are [[0.001 0.001 0.033 0.022 0.    0.062]]
The mean of the samples was -0.772
Iteration 293
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.578]
The next parameters to simulate from are [[0.143 0.55  0.027 0.01  0.018 0.025]]
The mean of the samples was -0.572
Iteration 294
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.578]
The next parameters to simulate from are [[0.031 0.804 0.006 0.01  0.006 0.028]]
The mean of the samples was -0.387
Iteration 295
Acquisition function convergence reached at iteration 72.
The final EI loss was -0.002 with predicted mean of [-1.747]
The next parameters to simulate from are [[0.055 0.387 0.005 0.01  0.017 0.023]]
The mean of the samples was -2.026
Iteration 296
Acquisition function convergence reached at iteration 770.
The final EI loss was -0.003 with predicted mean of [-1.489]
The next parameters to simulate from are [[0.406 0.001 0.011 0.011 0.025 0.019]]
The mean of the samples was -1.581
Iteration 297
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.215]
The next parameters to simulate from are [[0.049 0.244 0.006 0.04  0.034 0.039]]
The mean of the samples was 1.412
Iteration 298
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.879]
The next parameters to simulate from are [[0.771 0.42  0.003 0.042 0.029 0.024]]
The mean of the samples was 0.983
Iteration 299
Acquisition function convergence reached at iteration 90.
The final EI loss was -0.001 with predicted mean of [-0.648]
The next parameters to simulate from are [[0.001 0.995 0.028 0.011 0.047 0.036]]
The mean of the samples was -0.635
Iteration 300
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.185]
The next parameters to simulate from are [[0.725 0.42  0.009 0.01  0.05  0.024]]
The mean of the samples was -0.184
```

```
Hyperparameter convergence reached at iteration 2157.
The minimum predicted mean of the observed indices is -1.801 at the point
[0.049 0.42  0.006 0.01  0.018 0.024]
Trained parameters:
amplitude_champ:0 is 0.76

length_scales_champ:0 is [0.361 1.135 0.009 0.006 0.015 0.016]

observation_noise_variance_champ:0 is 0.067

bias_mean:0 is 0.89

dictionary saved successfully to file
Iteration 301
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.276]
The next parameters to simulate from are [[0.995 0.152 0.006 0.01  0.018 0.018]]
The mean of the samples was 0.362
Iteration 302
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.793]
The next parameters to simulate from are [[0.363 0.42  0.006 0.035 0.018 0.027]]
The mean of the samples was 0.808
Iteration 303
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.58]
The next parameters to simulate from are [[0.049 0.553 0.024 0.01  0.018 0.024]]
The mean of the samples was -0.619
Iteration 304
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.497]
The next parameters to simulate from are [[0.527 0.42  0.006 0.01  0.043 0.024]]
The mean of the samples was -0.551
Iteration 305
Acquisition function convergence reached at iteration 664.
The final EI loss was -0.001 with predicted mean of [-0.412]
The next parameters to simulate from are [[0.    0.001 0.024 0.022 0.    0.067]]
The mean of the samples was -0.396
Iteration 306
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.643]
The next parameters to simulate from are [[0.049 0.167 0.022 0.01  0.018 0.024]]
The mean of the samples was -0.627
```

```
Iteration 307
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.438]
The next parameters to simulate from are [[0.802 0.42  0.006 0.01  0.011 0.065]]
The mean of the samples was 0.43
Iteration 308
Acquisition function convergence reached at iteration 361.
The final EI loss was -0.0 with predicted mean of [0.24]
The next parameters to simulate from are [[0.999 0.002 0.033 0.025 0.05  0.   ]]
The mean of the samples was 0.742
Iteration 309
Acquisition function convergence reached at iteration 80.
The final EI loss was -0.001 with predicted mean of [-1.409]
The next parameters to simulate from are [[0.474 0.357 0.011 0.012 0.024 0.016]]
The mean of the samples was -1.548
Iteration 310
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.498]
The next parameters to simulate from are [[0.049 0.133 0.006 0.019 0.038 0.024]]
The mean of the samples was 0.336
Iteration 311
Acquisition function convergence reached at iteration 446.
The final EI loss was -0.001 with predicted mean of [-0.639]
The next parameters to simulate from are [[0.389 0.002 0.033 0.008 0.038 0.041]]
The mean of the samples was -0.69
Iteration 312
Acquisition function convergence reached at iteration 79.
The final EI loss was -0.002 with predicted mean of [-1.766]
The next parameters to simulate from are [[0.051 0.395 0.005 0.01  0.017 0.022]]
The mean of the samples was -1.933
Iteration 313
Acquisition function convergence reached at iteration 98.
The final EI loss was -0.0 with predicted mean of [-0.306]
The next parameters to simulate from are [[0.002 0.999 0.033 0.017 0.04  0.027]]
The mean of the samples was -0.109
Iteration 314
Acquisition function convergence reached at iteration 106.
The final EI loss was -0.002 with predicted mean of [-1.772]
The next parameters to simulate from are [[0.044 0.385 0.005 0.01  0.017 0.022]]
The mean of the samples was -1.732
Iteration 315
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.088]
```

```
The next parameters to simulate from are [[0.004 0.42  0.005 0.01  0.038 0.042]]
The mean of the samples was 0.022
Iteration 316
Acquisition function convergence reached at iteration 78.
The final EI loss was -0.002 with predicted mean of [-1.739]
The next parameters to simulate from are [[0.101 0.341 0.007 0.01  0.019 0.023]]
The mean of the samples was -1.792
Iteration 317
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.228]
The next parameters to simulate from are [[0.049 0.42  0.006 0.043 0.018 0.039]]
The mean of the samples was 1.278
Iteration 318
Acquisition function convergence reached at iteration 377.
The final EI loss was -0.0 with predicted mean of [-0.554]
The next parameters to simulate from are [[0.    0.998 0.013 0.017 0.    0.042]]
The mean of the samples was -0.323
Iteration 319
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.263]
The next parameters to simulate from are [[0.512 0.42  0.006 0.044 0.018 0.005]]
The mean of the samples was 1.283
Iteration 320
Acquisition function convergence reached at iteration 131.
The final EI loss was -0.002 with predicted mean of [-1.595]
The next parameters to simulate from are [[0.21  0.201 0.009 0.011 0.022 0.022]]
The mean of the samples was -1.694
Hyperparameter convergence reached at iteration 2087.
The minimum predicted mean of the observed indices is -1.817 at the point
[0.049 0.42  0.006 0.01  0.018 0.024]
Iteration 321
Acquisition function convergence reached at iteration 378.
The final EI loss was -0.001 with predicted mean of [-0.706]
The next parameters to simulate from are [[0.    0.001 0.03  0.009 0.036 0.06 ]]
The mean of the samples was -0.816
Iteration 322
Acquisition function convergence reached at iteration 127.
The final EI loss was -0.002 with predicted mean of [-0.923]
The next parameters to simulate from are [[0.001 0.98  0.011 0.006 0.037 0.033]]
The mean of the samples was -0.945
Iteration 323
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.205]
```

```
The next parameters to simulate from are [[0.049 0.443 0.006 0.01  0.039 0.041]]
The mean of the samples was -0.212
Iteration 324
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.437]
The next parameters to simulate from are [[0.049 0.42  0.006 0.042 0.044 0.031]]
The mean of the samples was 1.533
Iteration 325
Acquisition function convergence reached at iteration 241.
The final EI loss was -0.0 with predicted mean of [0.155]
The next parameters to simulate from are [[0.996 0.002 0.033 0.024 0.017 0.038]]
The mean of the samples was 0.446
Iteration 326
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.273]
The next parameters to simulate from are [[0.985 0.42  0.006 0.01  0.049 0.03 ]]
The mean of the samples was 0.458
Iteration 327
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.485]
The next parameters to simulate from are [[0.049 0.42  0.029 0.011 0.018 0.024]]
The mean of the samples was -0.507
Iteration 328
Acquisition function convergence reached at iteration 135.
The final EI loss was -0.001 with predicted mean of [-0.421]
The next parameters to simulate from are [[0.29  0.003 0.028 0.02  0.012 0.066]]
The mean of the samples was -0.503
Iteration 329
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.289]
The next parameters to simulate from are [[0.89  0.42  0.006 0.01  0.018 0.024]]
The mean of the samples was 0.279
Iteration 330
Acquisition function convergence reached at iteration 298.
The final EI loss was -0.001 with predicted mean of [-0.786]
The next parameters to simulate from are [[0.001 0.998 0.027 0.008 0.034 0.047]]
The mean of the samples was -0.786
Iteration 331
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.102]
The next parameters to simulate from are [[0.599 0.421 0.013 0.01  0.031 0.001]]
The mean of the samples was 0.241
Iteration 332
```

```
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.423]
The next parameters to simulate from are [[0.049 0.821 0.006 0.01  0.018 0.054]]
The mean of the samples was -0.44
Iteration 333
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.492]
The next parameters to simulate from are [[0.049 0.42  0.006 0.01  0.023 0.053]]
The mean of the samples was -0.458
Iteration 334
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.912]
The next parameters to simulate from are [[0.315 0.625 0.006 0.04  0.018 0.039]]
The mean of the samples was 0.907
Iteration 335
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.146]
The next parameters to simulate from are [[0.699 0.98  0.02  0.01  0.018 0.036]]
The mean of the samples was 0.082
Iteration 336
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.19]
The next parameters to simulate from are [[0.049 0.42  0.002 0.044 0.018 0.037]]
The mean of the samples was 1.317
Iteration 337
Acquisition function convergence reached at iteration 124.
The final EI loss was -0.001 with predicted mean of [-0.872]
The next parameters to simulate from are [[0.352 0.008 0.027 0.008 0.05  0.031]]
The mean of the samples was -0.87
Iteration 338
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.414]
The next parameters to simulate from are [[0.049 0.42  0.006 0.043 0.023 0.024]]
The mean of the samples was 1.397
Iteration 339
Acquisition function convergence reached at iteration 383.
The final EI loss was -0.0 with predicted mean of [0.254]
The next parameters to simulate from are [[0.    0.002 0.033 0.025 0.05  0.   ]]
The mean of the samples was 0.99
Iteration 340
Acquisition function convergence reached at iteration 308.
The final EI loss was -0.0 with predicted mean of [-1.418]
The next parameters to simulate from are [[0.    0.999 0.006 0.012 0.013 0.02 ]]
```

```
The mean of the samples was -1.528
Hyperparameter convergence reached at iteration 2174.
The minimum predicted mean of the observed indices is -1.815 at the point
[0.049 0.42  0.006 0.01  0.018 0.024]
Iteration 341
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.504]
The next parameters to simulate from are [[0.049 0.303 0.009 0.01  0.018 0.056]]
The mean of the samples was -0.44
Iteration 342
Acquisition function convergence reached at iteration 444.
The final EI loss was -0.002 with predicted mean of [-1.664]
The next parameters to simulate from are [[0.094 0.001 0.006 0.011 0.018 0.023]]
The mean of the samples was -1.881
Iteration 343
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.378]
The next parameters to simulate from are [[0.874 0.626 0.006 0.01  0.018 0.039]]
The mean of the samples was 0.374
Iteration 344
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.047]
The next parameters to simulate from are [[0.745 0.42  0.019 0.01  0.01  0.024]]
The mean of the samples was -0.058
Iteration 345
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.322]
The next parameters to simulate from are [[0.901 0.431 0.006 0.01  0.015 0.024]]
The mean of the samples was 0.294
Iteration 346
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.686]
The next parameters to simulate from are [[0.049 0.42  0.006 0.01  0.018 0.044]]
The mean of the samples was -0.657
Iteration 347
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.25]
The next parameters to simulate from are [[0.049 0.496 0.006 0.039 0.018 0.024]]
The mean of the samples was 1.246
Iteration 348
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.247]
The next parameters to simulate from are [[0.257 0.601 0.013 0.01  0.018 0.053]]
```

```
The mean of the samples was -0.208
Iteration 349
Acquisition function convergence reached at iteration 473.
The final EI loss was -0.001 with predicted mean of [-0.653]
The next parameters to simulate from are [[0.468 0.002 0.026 0.019 0.    0.026]]
The mean of the samples was -0.738
Iteration 350
Acquisition function convergence reached at iteration 98.
The final EI loss was -0.0 with predicted mean of [-0.526]
The next parameters to simulate from are [[0.002 0.011 0.026 0.012 0.    0.031]]
The mean of the samples was -0.507
Trained parameters:
amplitude_champ:0 is 0.768

length_scales_champ:0 is [0.368 1.084 0.008 0.006 0.016 0.017]

observation_noise_variance_champ:0 is 0.063

bias_mean:0 is 0.872

dictionary saved successfully to file
Iteration 351
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.257]
The next parameters to simulate from are [[0.02  0.42  0.006 0.004 0.018 0.024]]
The mean of the samples was -0.24
Iteration 352
Acquisition function convergence reached at iteration 282.
The final EI loss was -0.0 with predicted mean of [0.018]
The next parameters to simulate from are [[0.    0.997 0.014 0.028 0.    0.067]]
The mean of the samples was -0.17
Iteration 353
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.877]
The next parameters to simulate from are [[0.049 0.42  0.001 0.01  0.018 0.024]]
The mean of the samples was -0.721
Iteration 354
Acquisition function convergence reached at iteration 100.
The final EI loss was -0.002 with predicted mean of [-1.409]
The next parameters to simulate from are [[0.509 0.327 0.01  0.014 0.013 0.012]]
The mean of the samples was -1.306
Iteration 355
Acquisition function convergence reached at iteration 2.
```

```
The final EI loss was -0.0 with predicted mean of [-0.072]
The next parameters to simulate from are [[0.306 0.204 0.032 0.015 0.018 0.009]]
The mean of the samples was 0.008
Iteration 356
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.358]
The next parameters to simulate from are [[0.049 0.42  0.005 0.04  0.018 0.016]]
The mean of the samples was 1.37
Iteration 357
Acquisition function convergence reached at iteration 73.
The final EI loss was -0.0 with predicted mean of [-0.566]
The next parameters to simulate from are [[0.592 0.993 0.011 0.02  0.008 0.013]]
The mean of the samples was -0.617
Iteration 358
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.024]
The next parameters to simulate from are [[0.476 0.42  0.025 0.04  0.019 0.024]]
The mean of the samples was 0.884
Iteration 359
Acquisition function convergence reached at iteration 454.
The final EI loss was -0.001 with predicted mean of [-0.645]
The next parameters to simulate from are [[0.001 0.001 0.029 0.011 0.019 0.067]]
The mean of the samples was -0.599
Iteration 360
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 [[0.049 0.245 0.001 0.044 0.01  0.024]]
The mean of the samples was 1.359
Hyperparameter convergence reached at iteration 2212.
The minimum predicted mean of the observed indices is -1.809 at the point
[0.049 0.42  0.006 0.01  0.018 0.024]
Iteration 361
Acquisition function convergence reached at iteration 444.
The final EI loss was -0.0 with predicted mean of [0.226]
The next parameters to simulate from are [[0.998 0.998 0.018 0.023 0.    0.067]]
The mean of the samples was 0.397
Iteration 362
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.982]
The next parameters to simulate from are [[0.302 0.659 0.013 0.034 0.043 0.045]]
The mean of the samples was 1.009
Iteration 363
Acquisition function convergence reached at iteration 2.
```

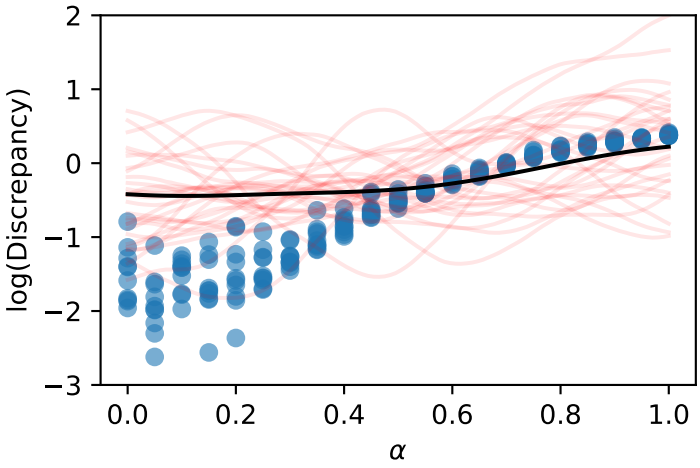
```
The final EI loss was -0.0 with predicted mean of [-0.837]
The next parameters to simulate from are [[0.422 0.307 0.018 0.01  0.018 0.024]]
The mean of the samples was -0.775
Iteration 364
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.336]
The next parameters to simulate from are [[0.007 0.42  0.028 0.045 0.025 0.048]]
The mean of the samples was 1.436
Iteration 365
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.278]
The next parameters to simulate from are [[0.587 0.42  0.006 0.01  0.018 0.024]]
The mean of the samples was -0.244
Iteration 366
Acquisition function convergence reached at iteration 321.
The final EI loss was -0.001 with predicted mean of [-0.738]
The next parameters to simulate from are [[0.497 0.998 0.027 0.017 0.034 0.023]]
The mean of the samples was -0.951
Iteration 367
Acquisition function convergence reached at iteration 129.
The final EI loss was -0.002 with predicted mean of [-1.141]
The next parameters to simulate from are [[0.126 0.002 0.012 0.007 0.036 0.028]]
The mean of the samples was -1.114
Iteration 368
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.78]
The next parameters to simulate from are [[0.811 0.42  0.006 0.041 0.018 0.024]]
The mean of the samples was 0.72
Iteration 369
Acquisition function convergence reached at iteration 406.
The final EI loss was -0.0 with predicted mean of [0.198]
The next parameters to simulate from are [[0.001 0.001 0.026 0.    0.05  0.   ]]
The mean of the samples was 0.527
Iteration 370
Acquisition function convergence reached at iteration 417.
The final EI loss was -0.001 with predicted mean of [-0.117]
The next parameters to simulate from are [[0.002 0.002 0.033 0.03  0.    0.067]]
The mean of the samples was -0.33
Iteration 371
Acquisition function convergence reached at iteration 333.
The final EI loss was -0.0 with predicted mean of [0.227]
The next parameters to simulate from are [[0.    0.997 0.022 0.    0.    0.033]]
The mean of the samples was 0.43
```

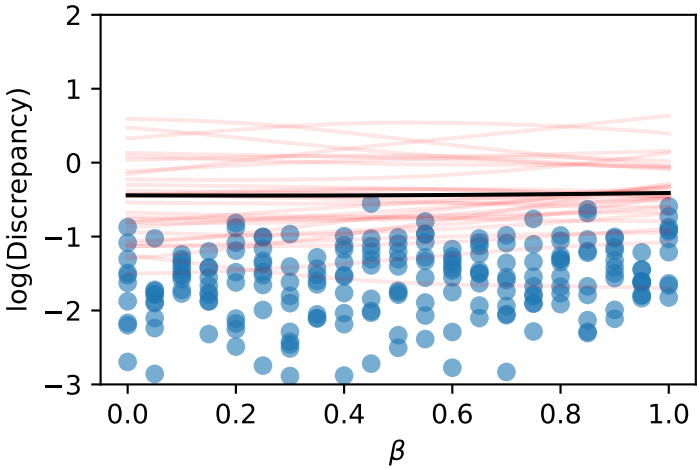
```
Iteration 372
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.585]
The next parameters to simulate from are [[0.049 0.129 0.006 0.024 0.038 0.024]]
The mean of the samples was 0.69
Iteration 373
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.556]
The next parameters to simulate from are [[0.017 0.42  0.006 0.022 0.033 0.034]]
The mean of the samples was 0.614
Iteration 374
Acquisition function convergence reached at iteration 612.
The final EI loss was -0.002 with predicted mean of [-0.683]
The next parameters to simulate from are [[0.179 1.    0.033 0.021 0.    0.054]]
The mean of the samples was -0.779
Iteration 375
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.243]
The next parameters to simulate from are [[0.049 0.42  0.006 0.01  0.018 0.002]]
The mean of the samples was 0.27
Iteration 376
Acquisition function convergence reached at iteration 713.
The final EI loss was -0.002 with predicted mean of [-1.729]
The next parameters to simulate from are [[0.085 0.001 0.006 0.011 0.018 0.023]]
The mean of the samples was -1.743
Iteration 377
Acquisition function convergence reached at iteration 340.
The final EI loss was -0.001 with predicted mean of [-1.032]
The next parameters to simulate from are [[0.    0.001 0.011 0.006 0.03  0.042]]
The mean of the samples was -0.839
Iteration 378
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.574]
The next parameters to simulate from are [[0.822 0.42  0.006 0.021 0.04  0.024]]
The mean of the samples was 0.312
Iteration 379
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.046]
The next parameters to simulate from are [[0.148 0.42  0.001 0.01  0.018 0.044]]
The mean of the samples was -0.208
Iteration 380
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.674]
```

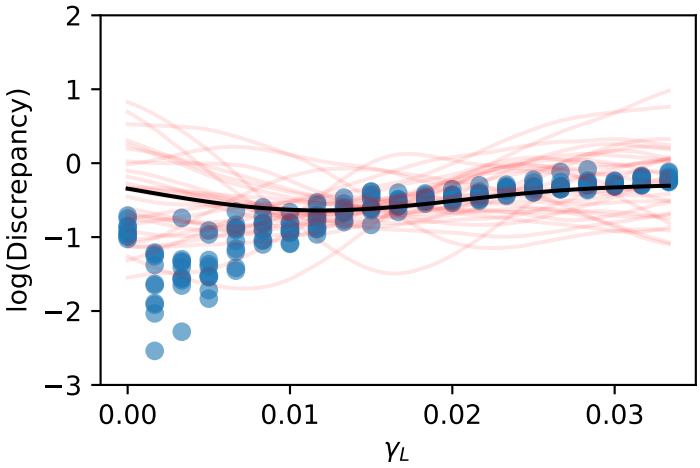
```
The next parameters to simulate from are [[0.049 0.697 0.031 0.01  0.018 0.049]]
The mean of the samples was -0.634
Hyperparameter convergence reached at iteration 2057.
The minimum predicted mean of the observed indices is -1.81 at the point
[0.049 0.42  0.006 0.01  0.018 0.024]
Iteration 381
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.089]
The next parameters to simulate from are [[0.049 0.406 0.006 0.003 0.018 0.024]]
The mean of the samples was -0.095
Iteration 382
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.118]
The next parameters to simulate from are [[0.136 0.42  0.014 0.01  0.018 0.006]]
The mean of the samples was -0.019
Iteration 383
Acquisition function convergence reached at iteration 168.
The final EI loss was -0.001 with predicted mean of [-1.658]
The next parameters to simulate from are [[0.    0.999 0.006 0.01  0.019 0.024]]
The mean of the samples was -1.752
Iteration 384
Acquisition function convergence reached at iteration 377.
The final EI loss was -0.0 with predicted mean of [-0.292]
The next parameters to simulate from are [[0.    0.998 0.006 0.003 0.05  0.027]]
The mean of the samples was -0.631
Iteration 385
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.887]
The next parameters to simulate from are [[0.049 0.868 0.006 0.014 0.021 0.024]]
The mean of the samples was -0.823
Iteration 386
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.703]
The next parameters to simulate from are [[0.049 0.386 0.006 0.01  0.033 0.032]]
The mean of the samples was -0.595
Iteration 387
Acquisition function convergence reached at iteration 91.
The final EI loss was -0.001 with predicted mean of [-1.788]
The next parameters to simulate from are [[0.09  0.299 0.006 0.01  0.019 0.023]]
The mean of the samples was -1.852
Iteration 388
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.27]
```

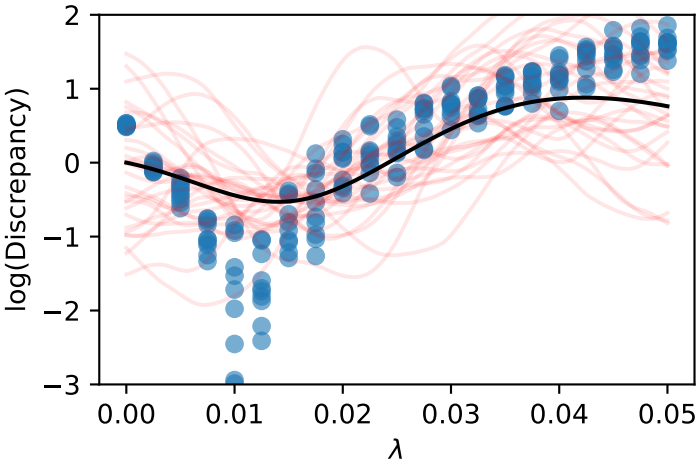
```
The next parameters to simulate from are [[0.049 0.42  0.006 0.01  0.032 0.054]]
The mean of the samples was -0.216
Iteration 389
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [0.859]
The next parameters to simulate from are [[0.715 0.42  0.006 0.036 0.033 0.024]]
The mean of the samples was 0.876
Iteration 390
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.491]
The next parameters to simulate from are [[0.049 0.42  0.006 0.01  0.043 0.024]]
The mean of the samples was -0.537
Iteration 391
Acquisition function convergence reached at iteration 44.
The final EI loss was -0.001 with predicted mean of [-1.795]
The next parameters to simulate from are [[0.088 0.301 0.006 0.01  0.019 0.024]]
The mean of the samples was -1.581
Iteration 392
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.319]
The next parameters to simulate from are [[0.67  0.051 0.006 0.049 0.039 0.012]]
The mean of the samples was 1.631
Iteration 393
Acquisition function convergence reached at iteration 544.
The final EI loss was -0.001 with predicted mean of [-1.66]
The next parameters to simulate from are [[0.    0.001 0.006 0.01  0.018 0.024]]
The mean of the samples was -1.624
Iteration 394
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.72]
The next parameters to simulate from are [[0.049 0.42  0.006 0.01  0.036 0.024]]
The mean of the samples was -0.68
Iteration 395
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [1.356]
The next parameters to simulate from are [[0.049 0.42  0.029 0.043 0.014 0.024]]
The mean of the samples was 1.421
Iteration 396
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.492]
The next parameters to simulate from are [[0.049 0.999 0.006 0.01  0.018 0.051]]
The mean of the samples was -0.473
Iteration 397
```

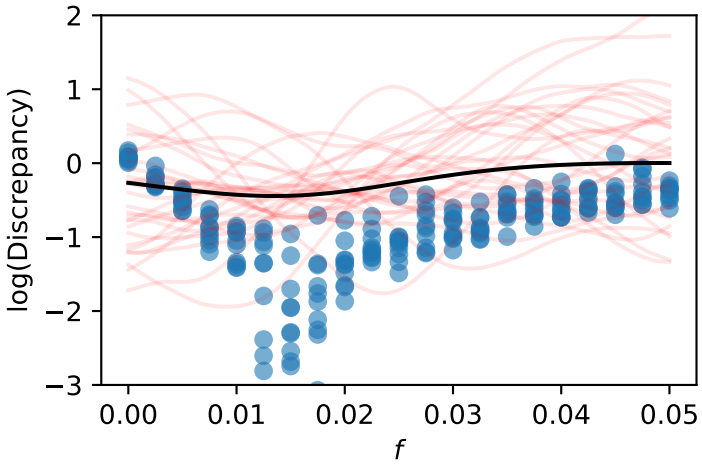
```
Acquisition function convergence reached at iteration 56.
The final EI loss was -0.0 with predicted mean of [-1.061]
The next parameters to simulate from are [[0.53  0.613 0.03  0.015 0.023 0.022]]
The mean of the samples was -1.007
Iteration 398
Acquisition function convergence reached at iteration 256.
The final EI loss was -0.0 with predicted mean of [-0.568]
The next parameters to simulate from are [[0.    0.001 0.015 0.007 0.05  0.037]]
The mean of the samples was -0.892
Iteration 399
Acquisition function convergence reached at iteration 2.
The final EI loss was -0.0 with predicted mean of [-0.613]
The next parameters to simulate from are [[0.049 0.918 0.006 0.01  0.038 0.024]]
The mean of the samples was -0.753
Iteration 400
Acquisition function convergence reached at iteration 119.
The final EI loss was -0.001 with predicted mean of [-0.995]
The next parameters to simulate from are [[0.425 0.002 0.027 0.012 0.038 0.035]]
The mean of the samples was -0.73
Hyperparameter convergence reached at iteration 2199.
The minimum predicted mean of the observed indices is -1.813 at the point
[0.049 0.42  0.006 0.01  0.018 0.024]
Trained parameters:
amplitude_champ:0 is 0.762

length_scales_champ:0 is [0.362 1.229 0.009 0.006 0.016 0.017]

observation_noise_variance_champ:0 is 0.064

bias_mean:0 is 0.881

dictionary saved successfully to file
```

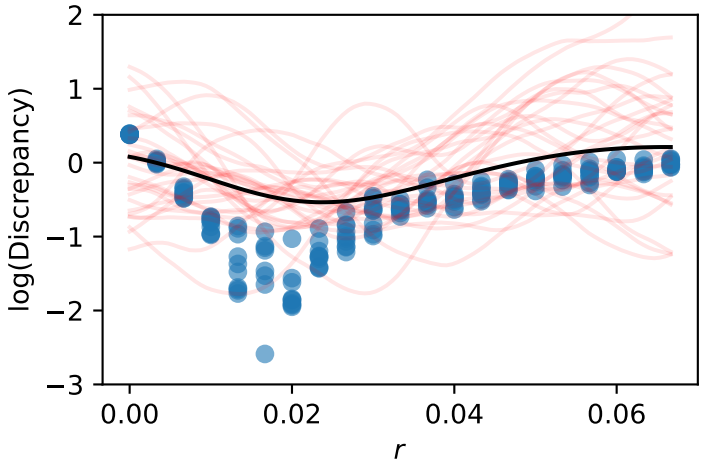

log(Discrepancy)
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
α

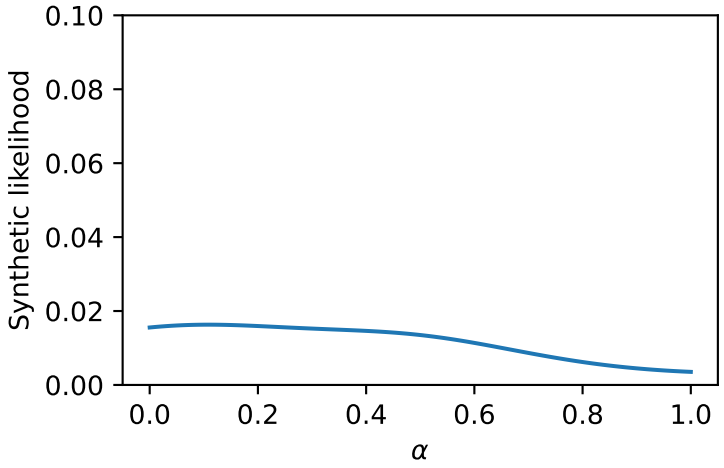

log(Discrepancy)
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
β

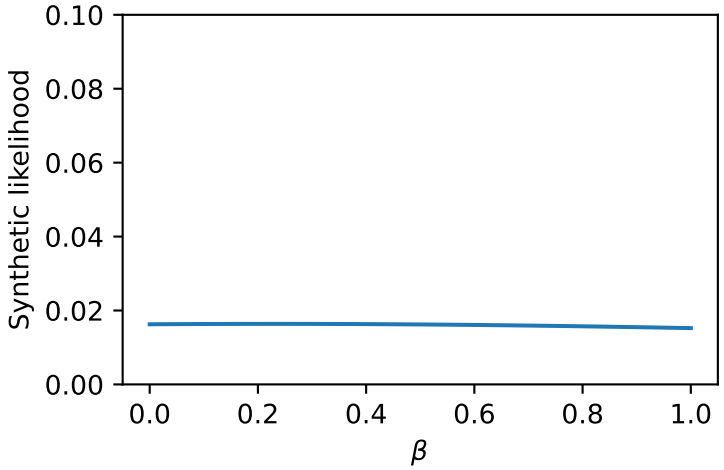

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
γ_L

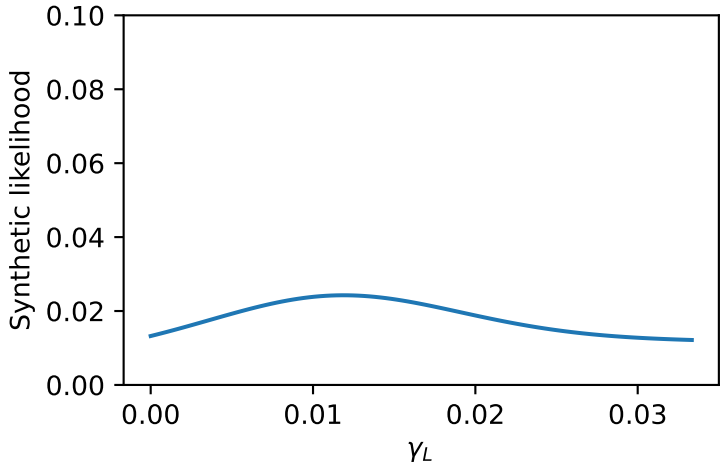

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
λ

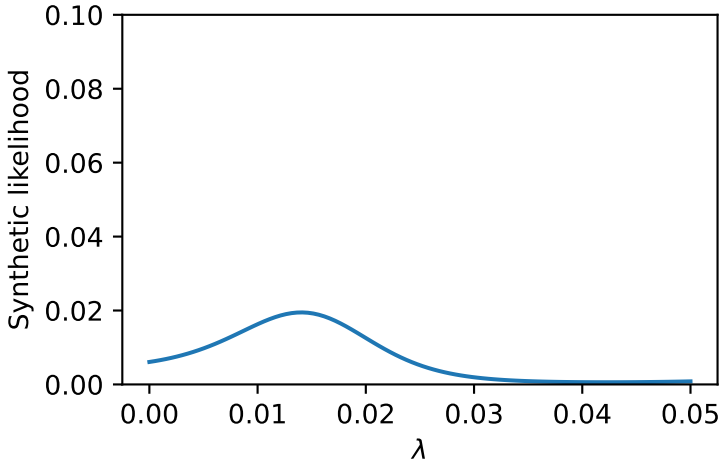

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
f

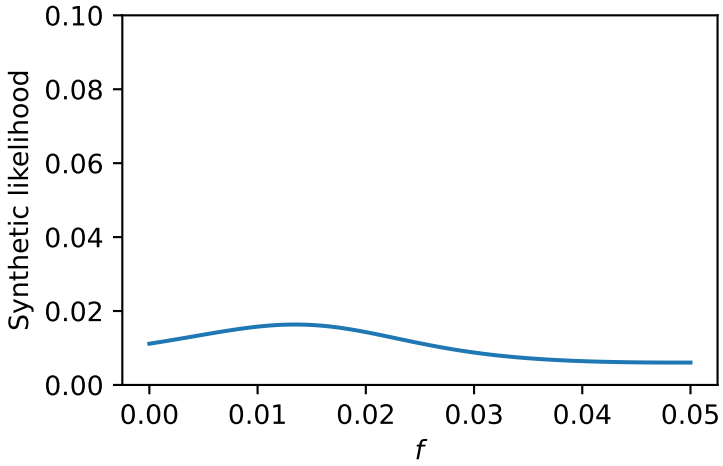

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.02
0.04
0.06
r

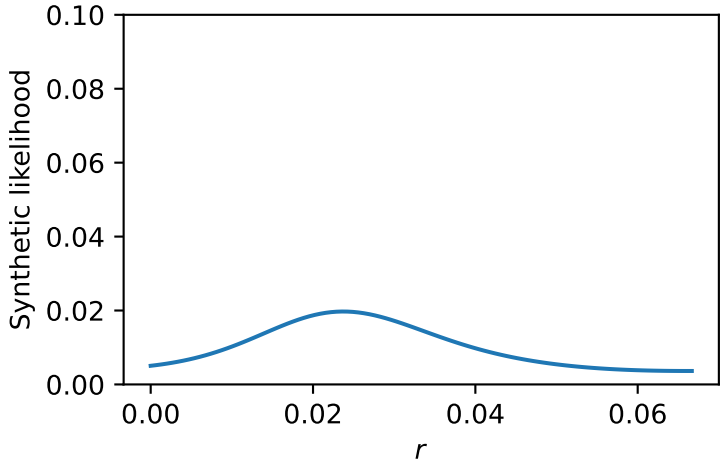

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
α

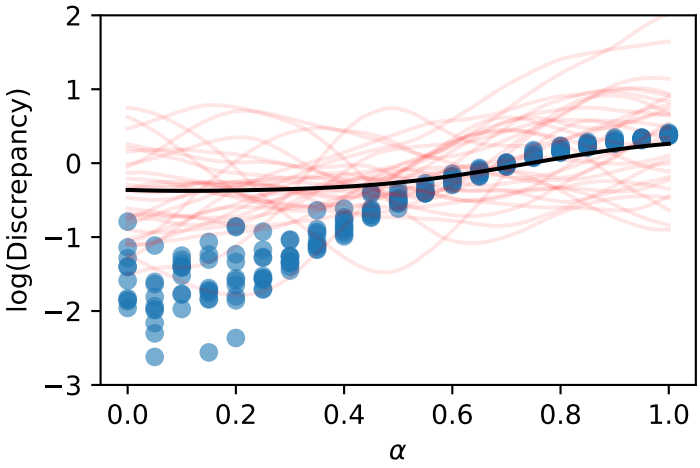

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
β

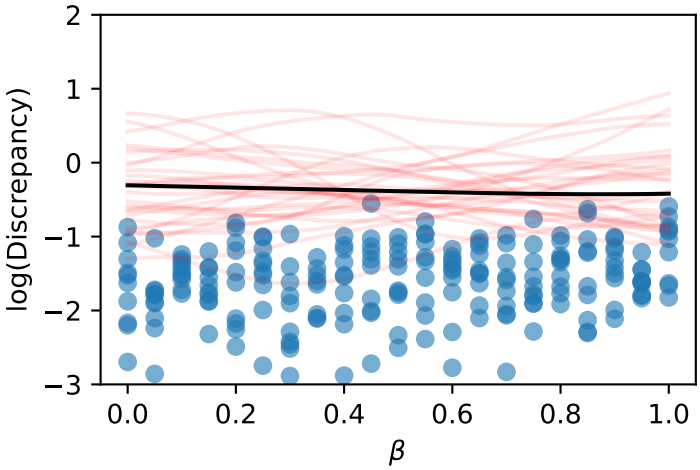

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
γ_L

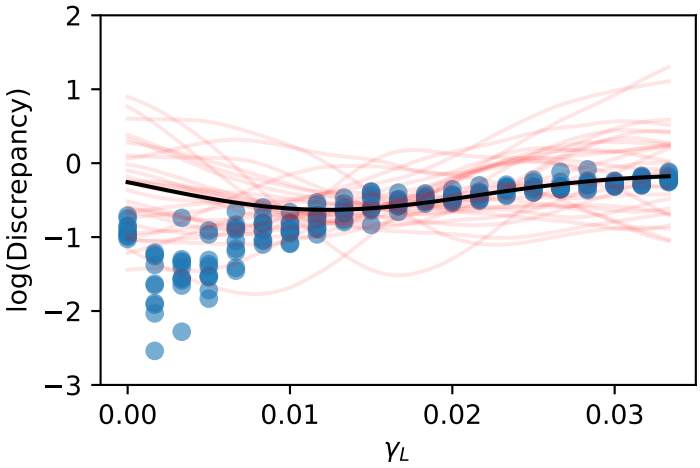

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
0.04
0.05
λ

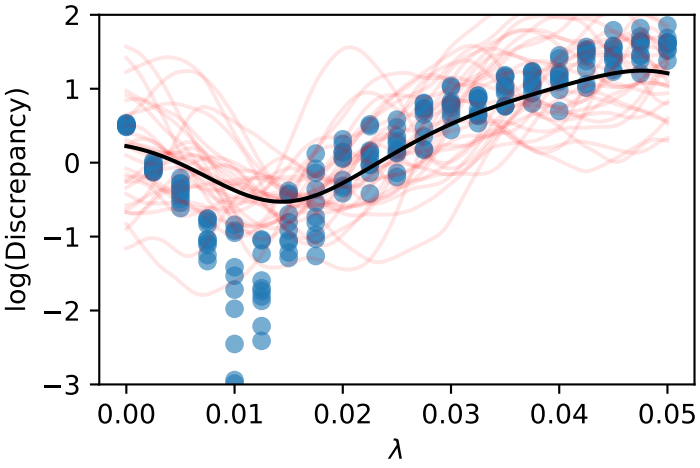

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
0.04
0.05
f

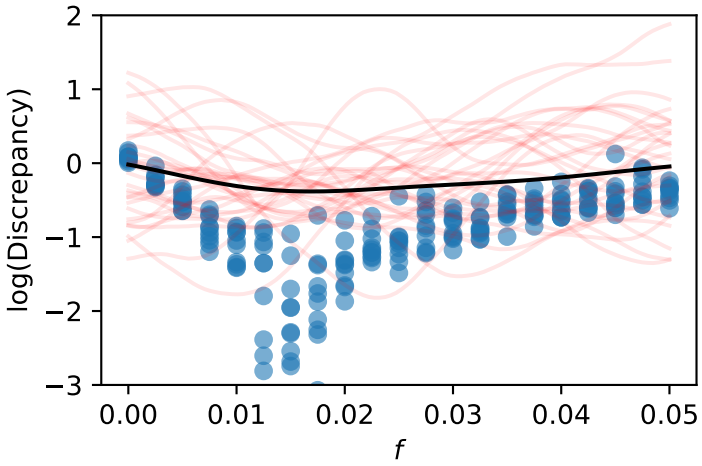

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.02
0.04
0.06
r

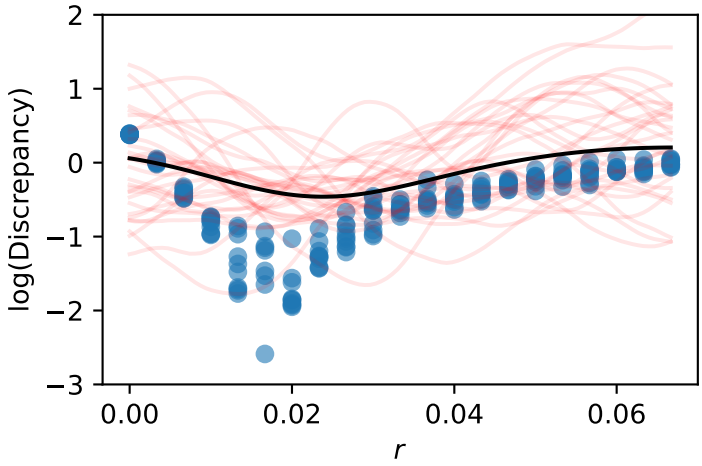

log(Discrepancy)
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
α

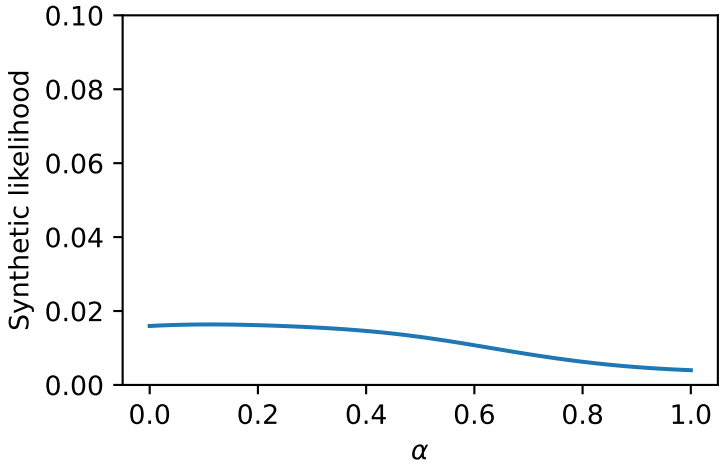

log(Discrepancy)
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
β

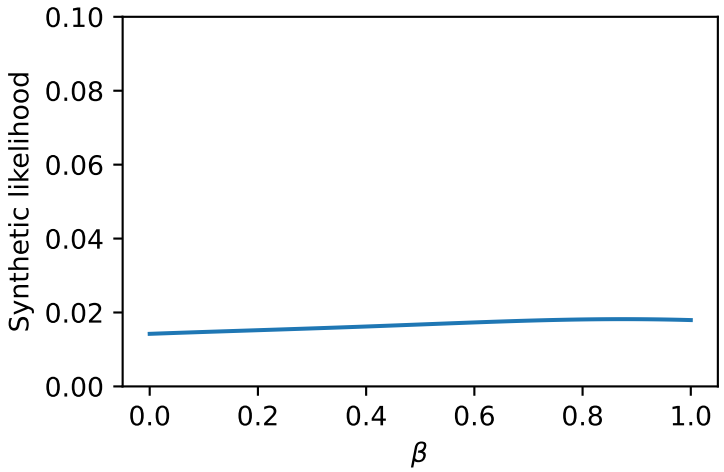

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
γ_L

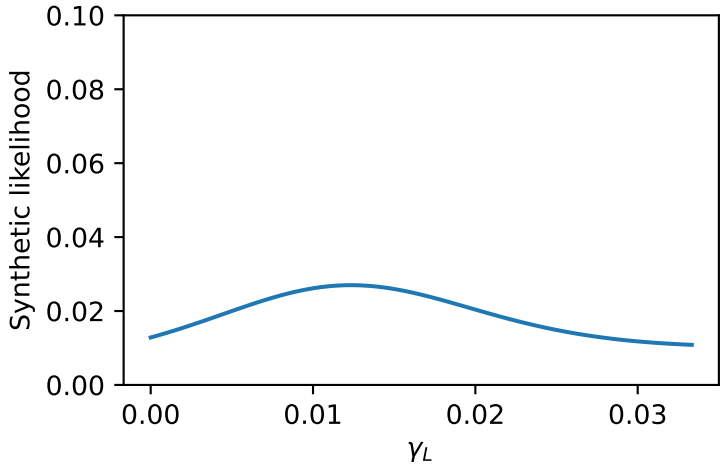

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
λ

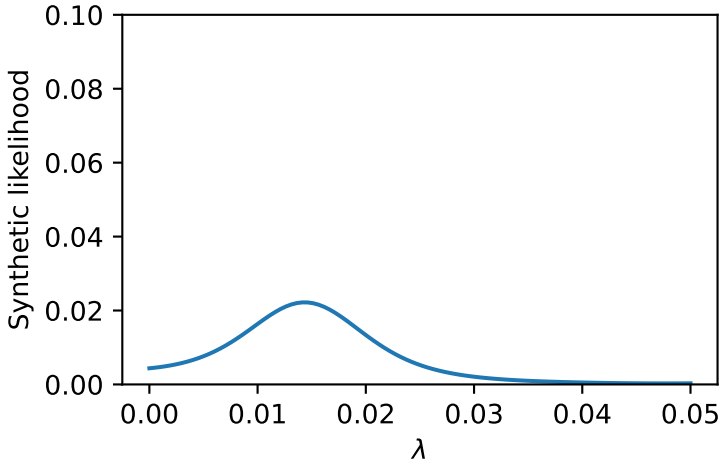

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
f

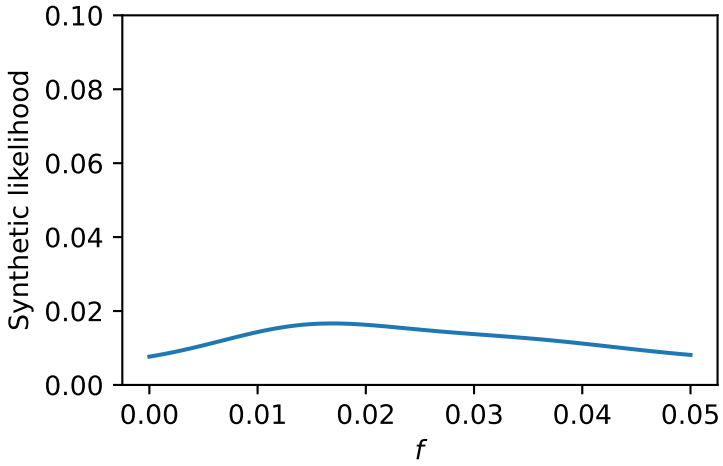

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.02
0.04
0.06
r

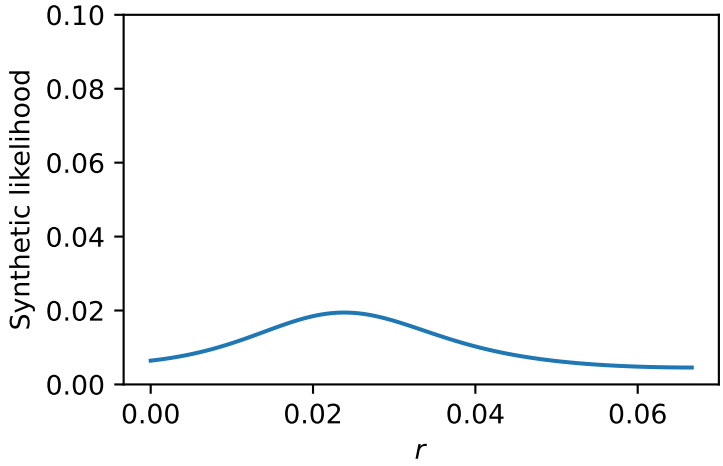

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
α

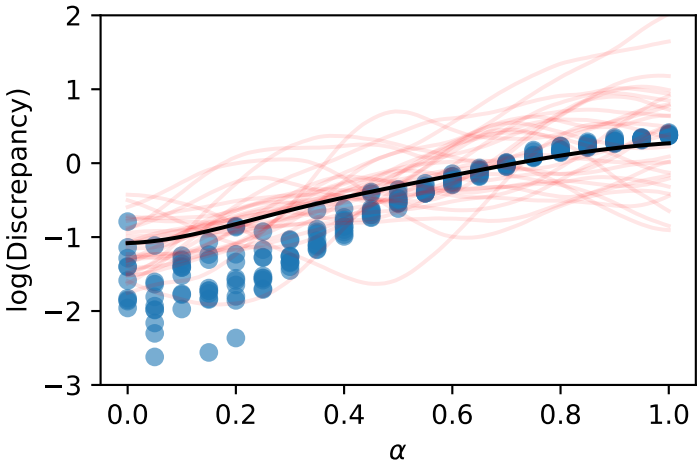

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
β

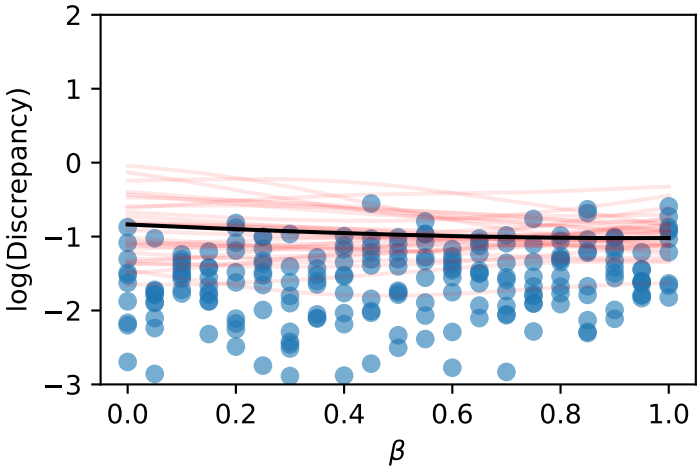

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
γ_L

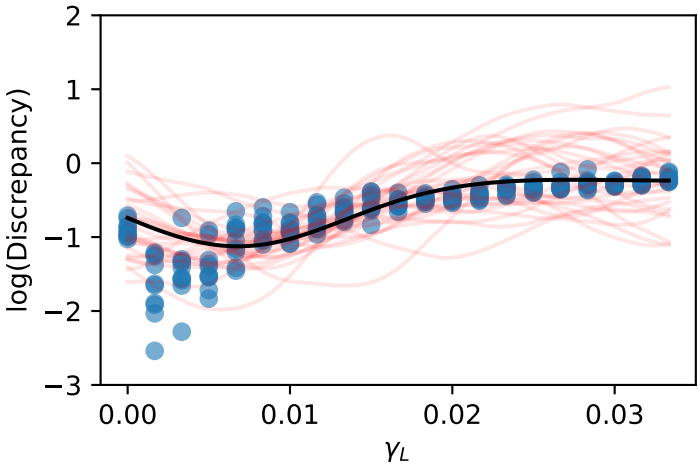

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
0.04
0.05
λ

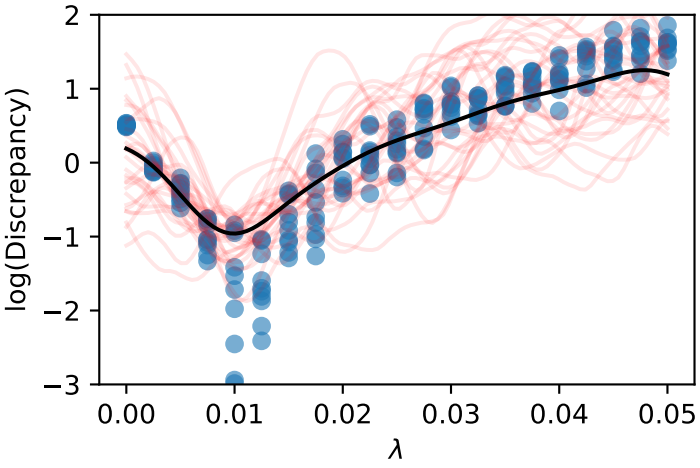

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
0.04
0.05
f

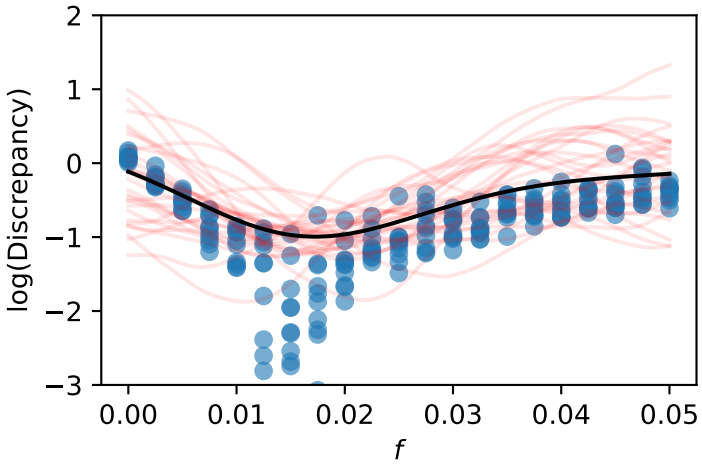

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.02
0.04
0.06
r

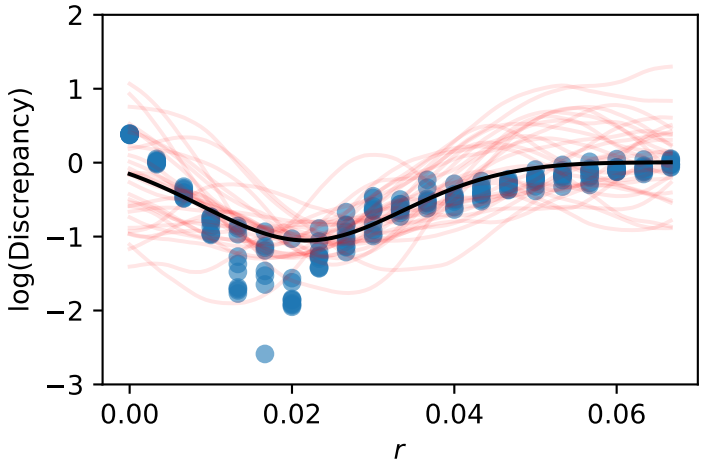

log(Discrepancy)
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
α

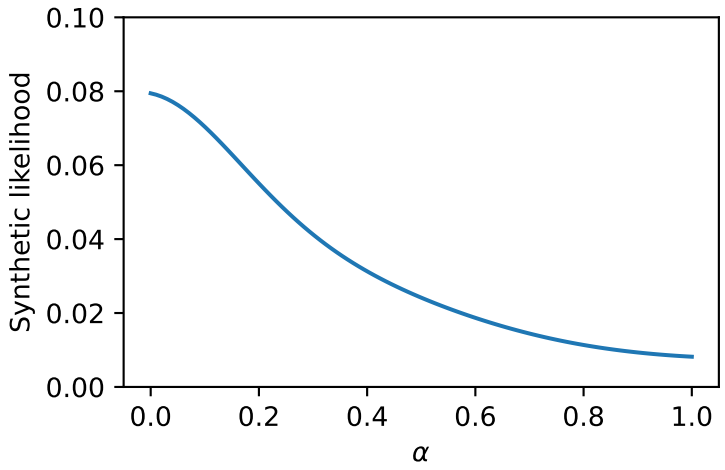

log(Discrepancy)
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
β

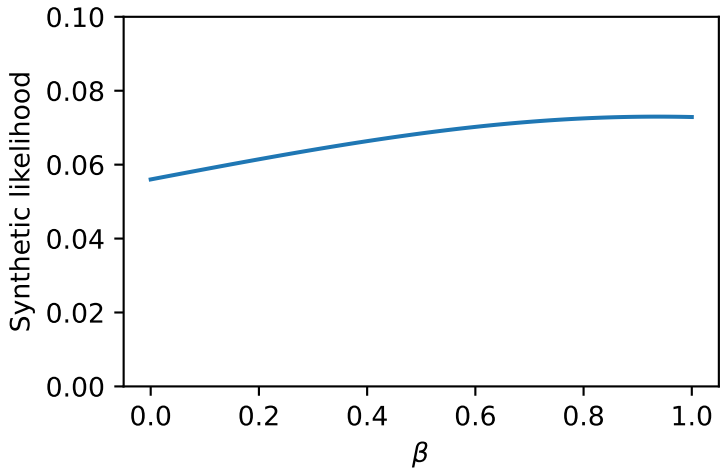

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
γ_L

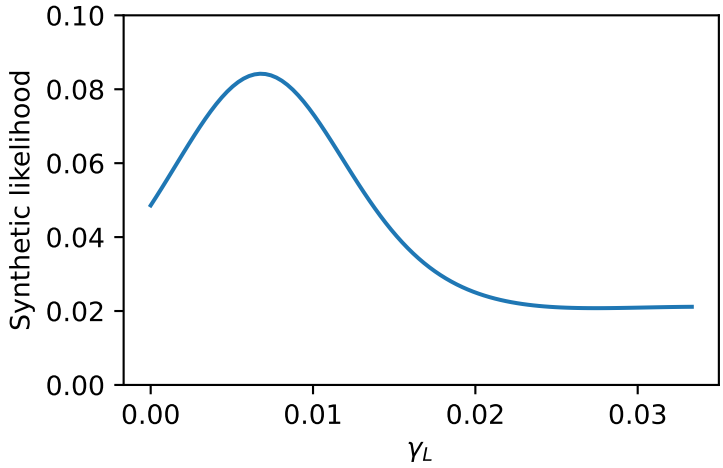

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
λ

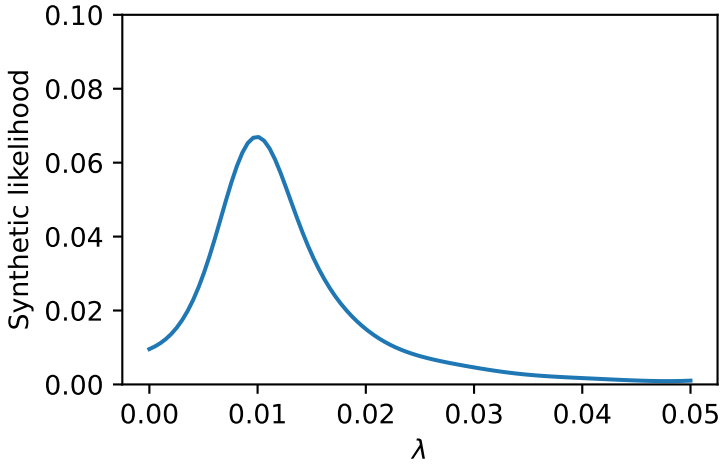

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
f

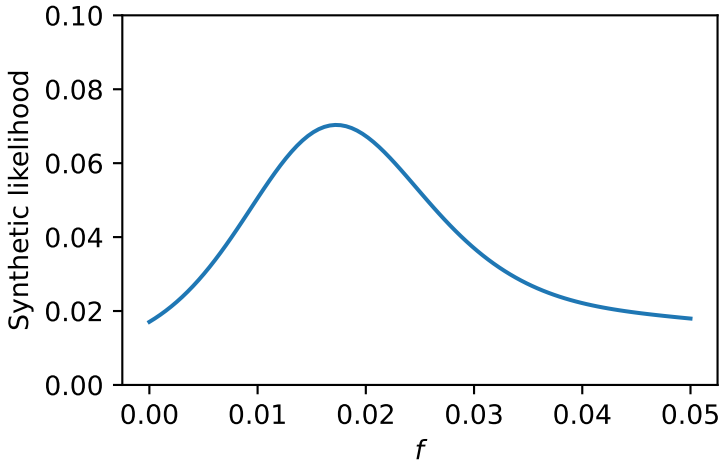

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.02
0.04
0.06
r

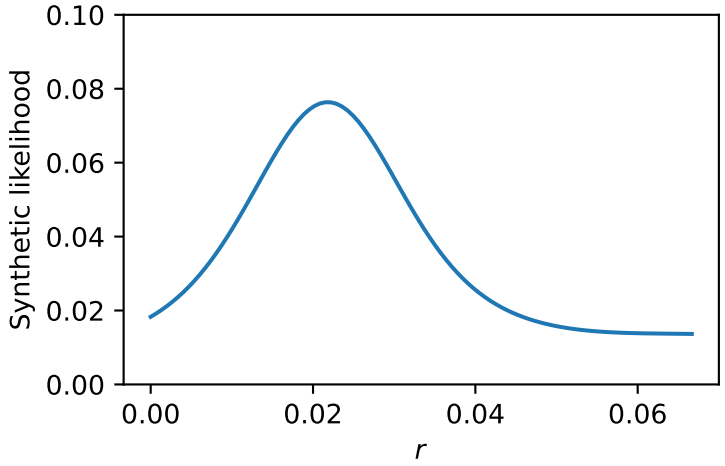

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
α

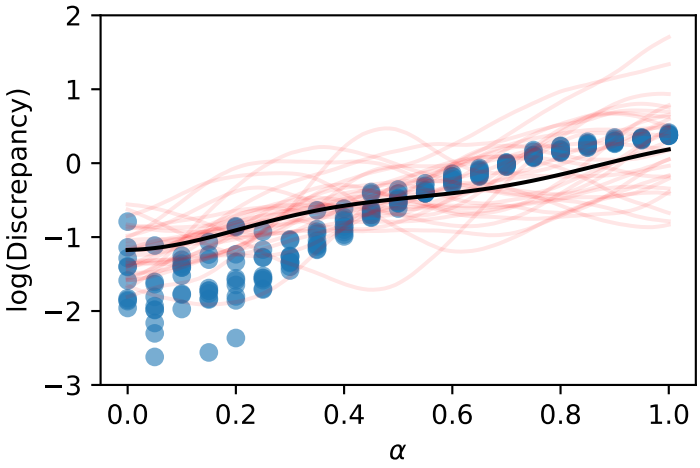

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
β

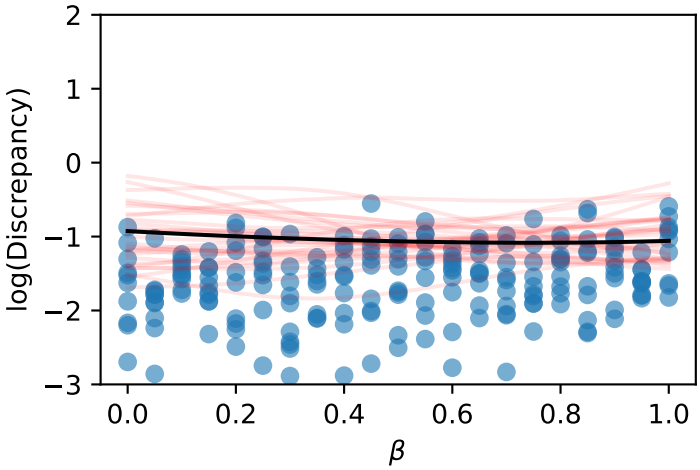

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
γ_L

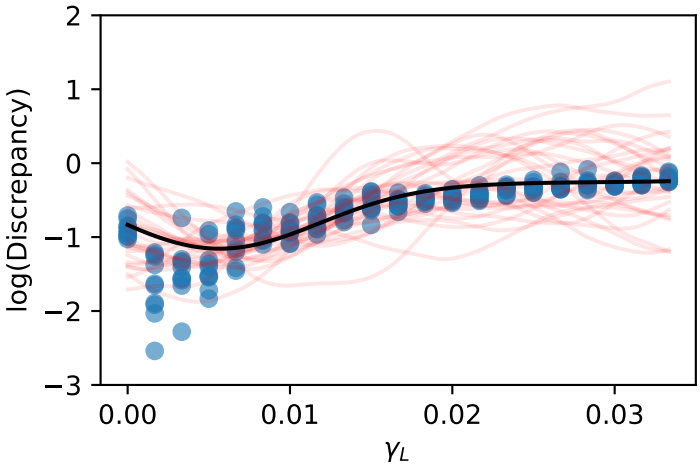

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
0.04
0.05
λ

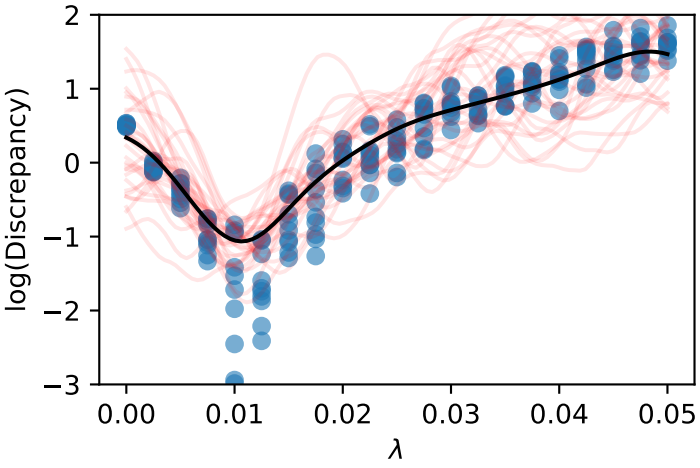

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
0.04
0.05
f

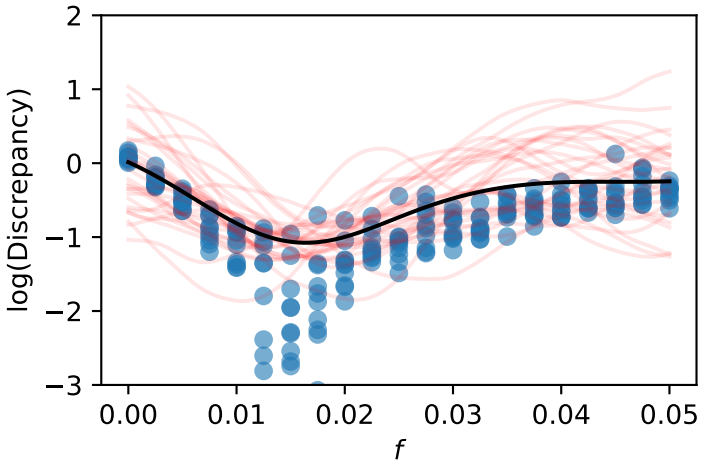

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.02
0.04
0.06
r

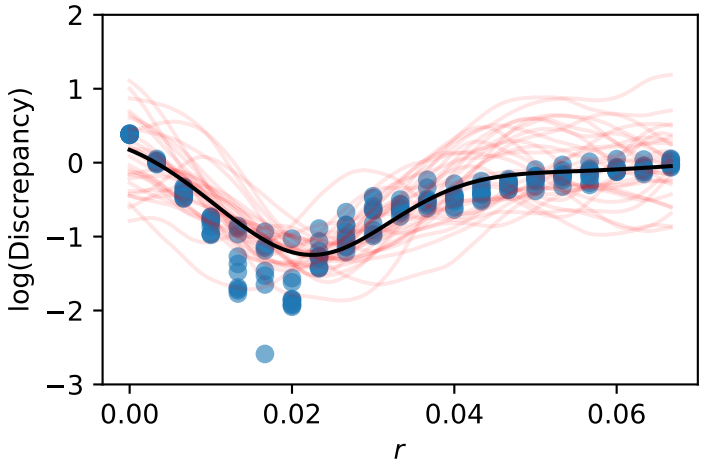

log(Discrepancy)
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
α

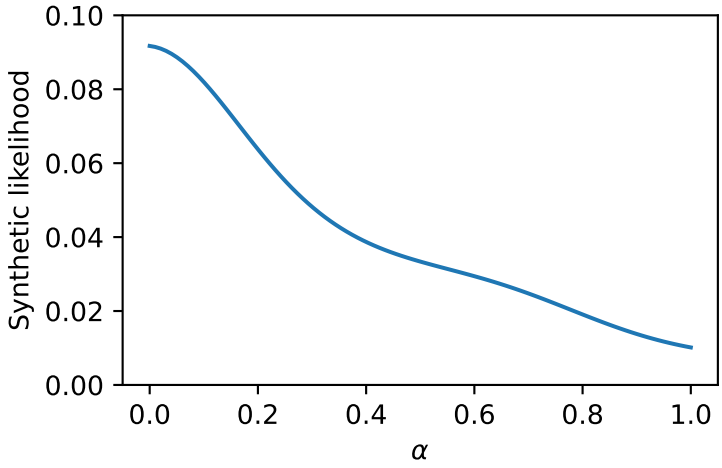

log(Discrepancy)
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
β

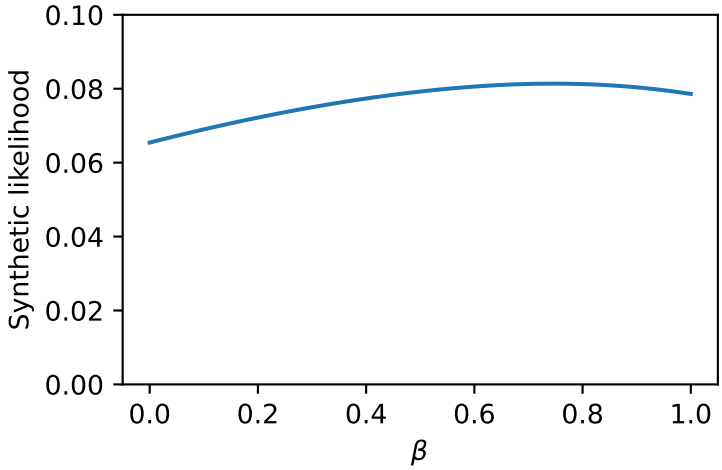

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
γ_L

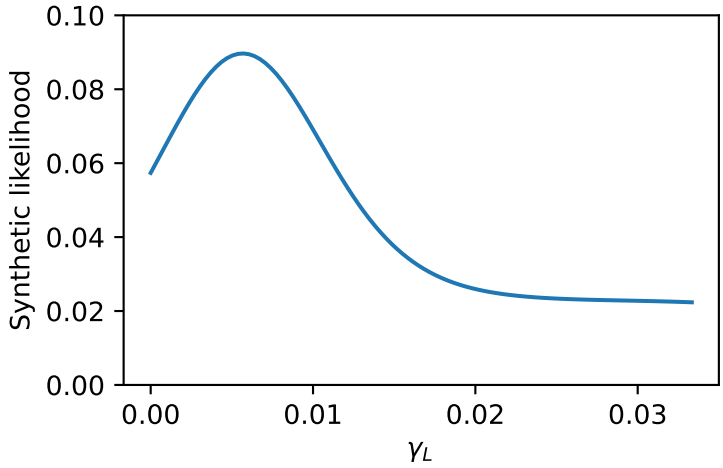

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
λ

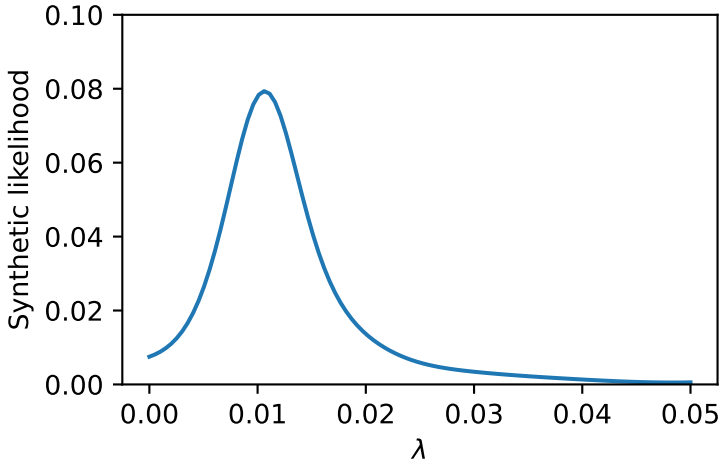

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
f

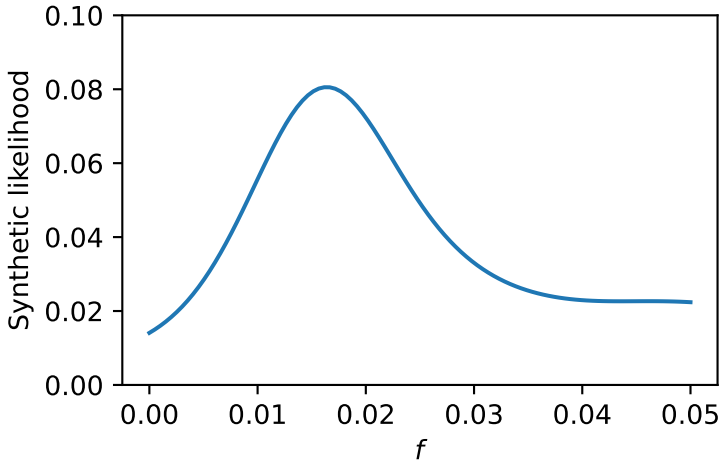

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.02
0.04
0.06
r

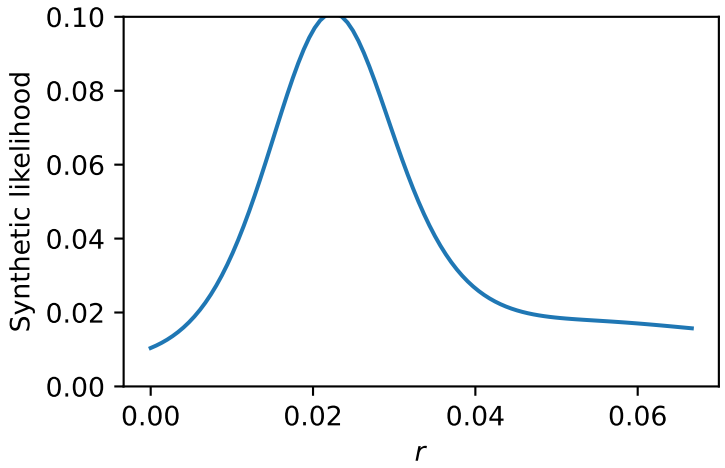

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
α

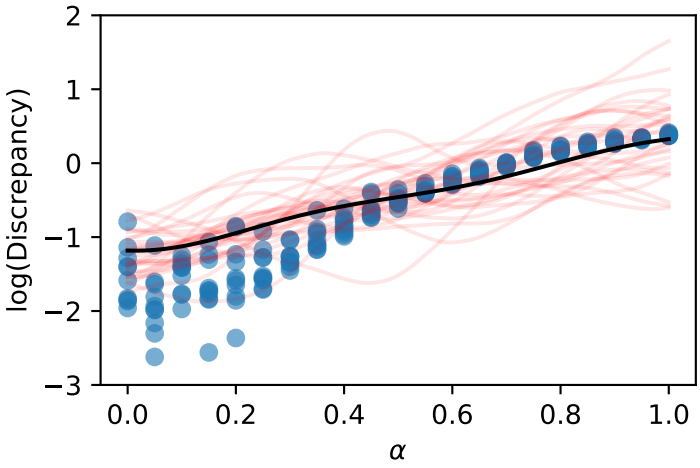

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
β

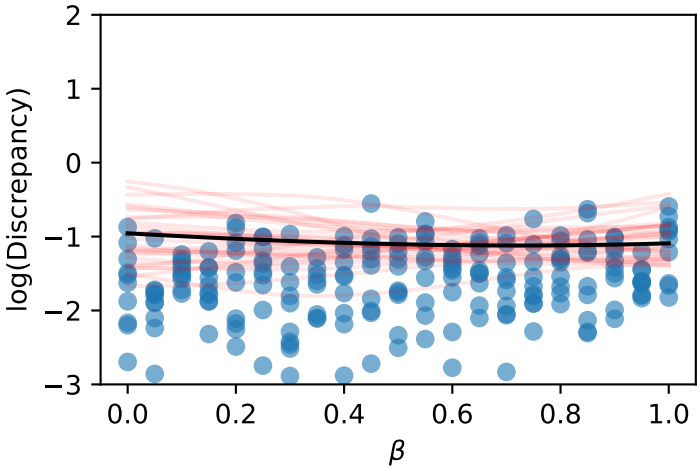

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
γ_L

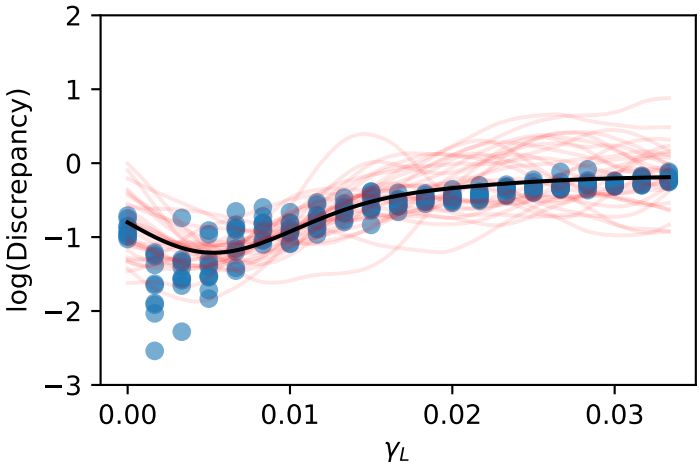

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
0.04
0.05
λ

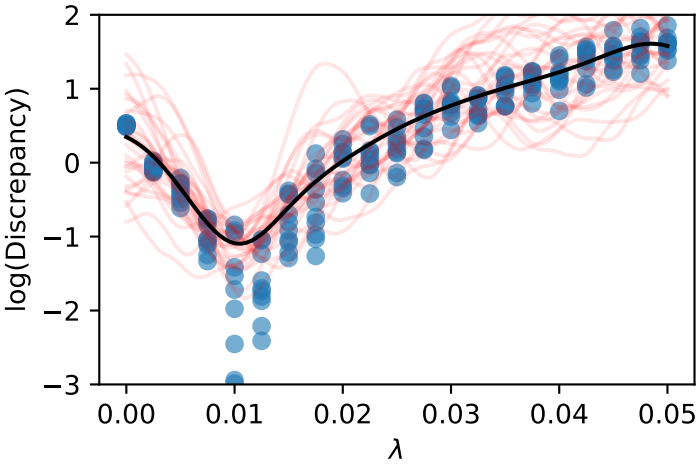

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
0.04
0.05
f

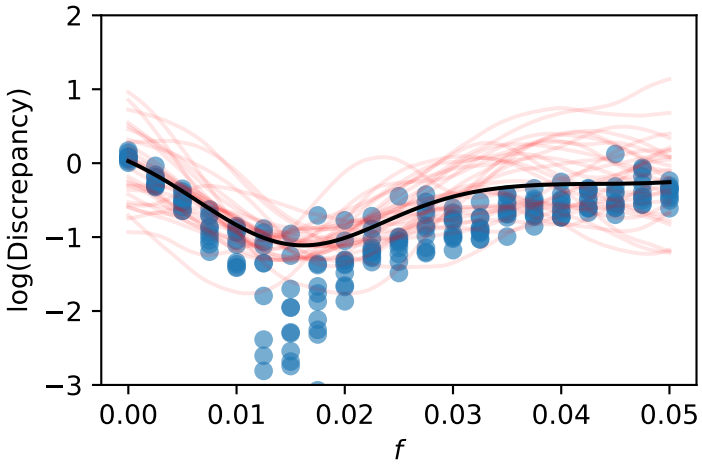

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.02
0.04
0.06
r

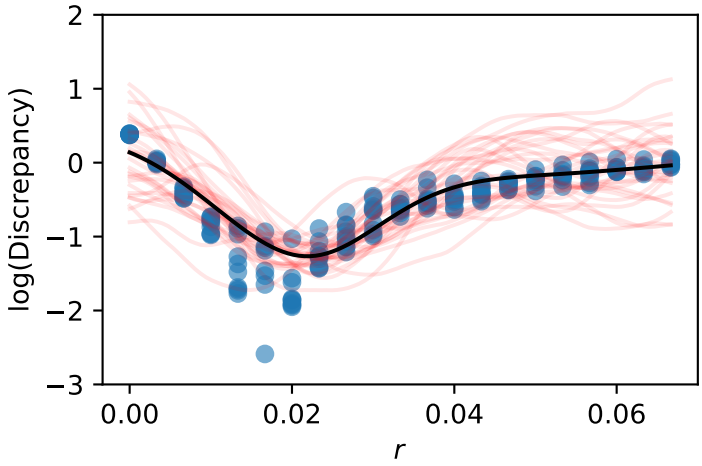

log(Discrepancy)
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
α

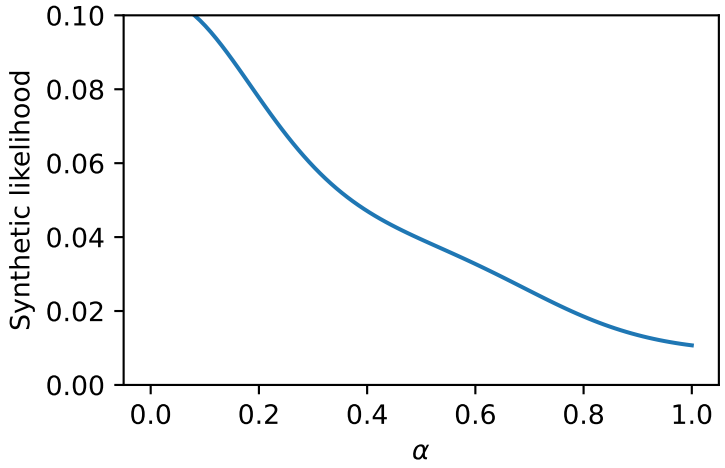

log(Discrepancy)
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
β

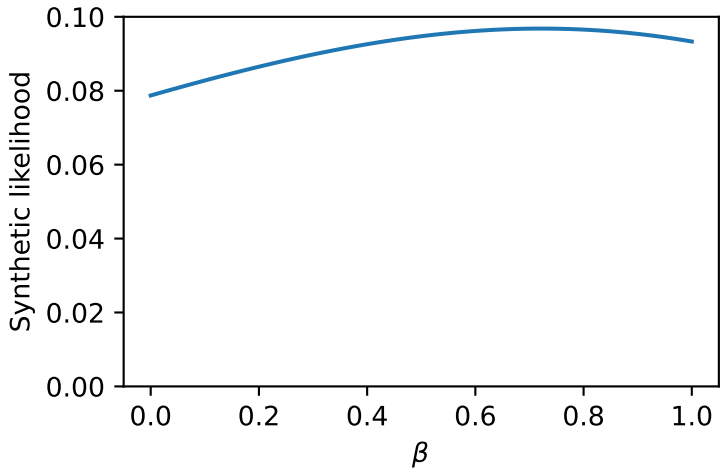

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
γ_L

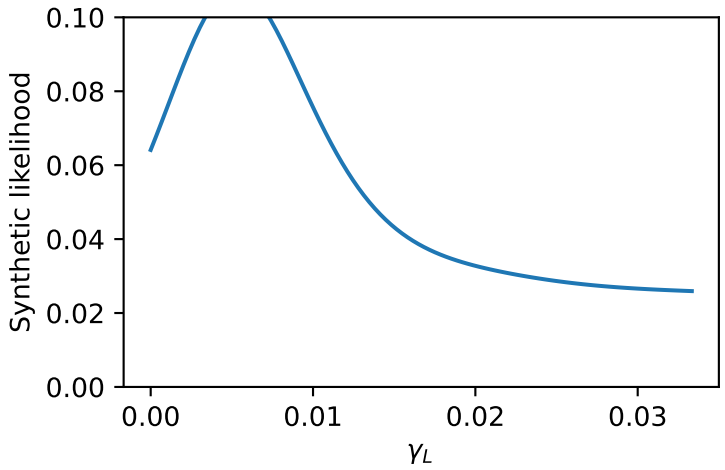

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
λ

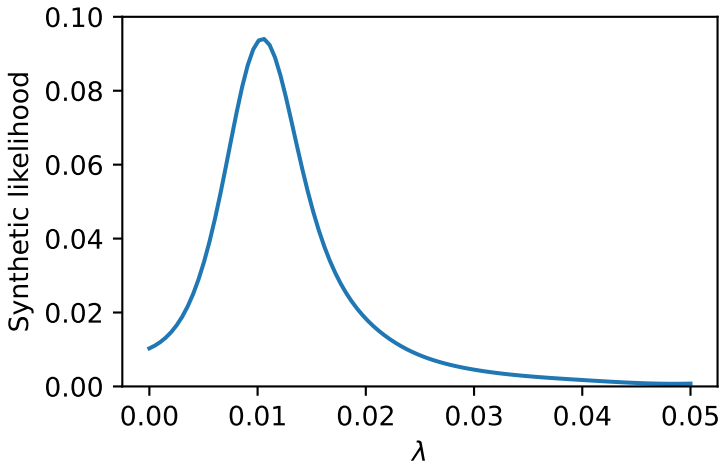

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
f

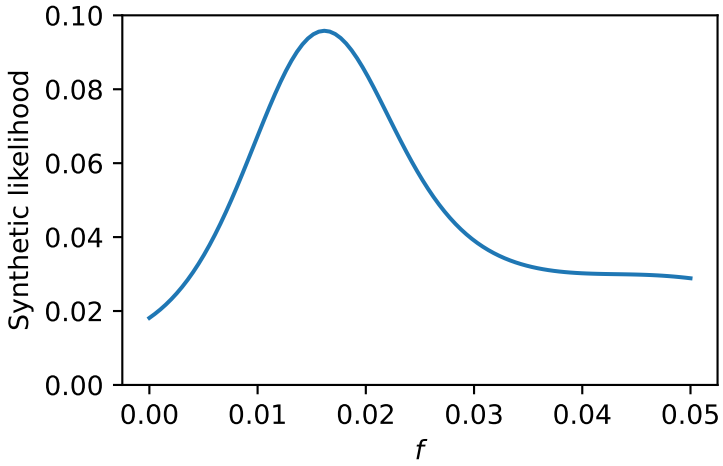

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.02
0.04
0.06
r

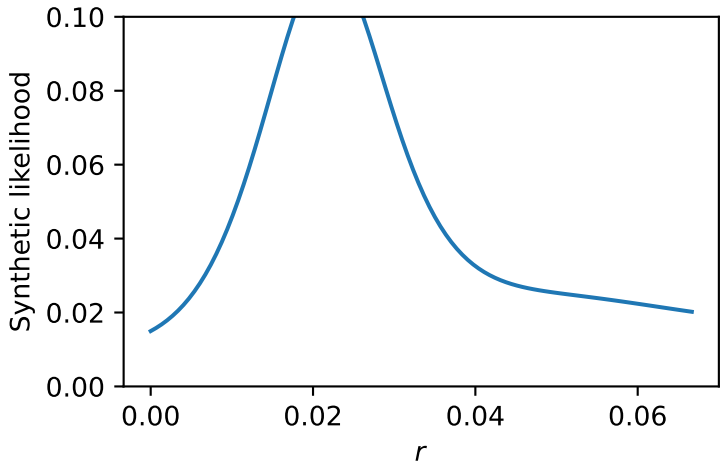

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
α

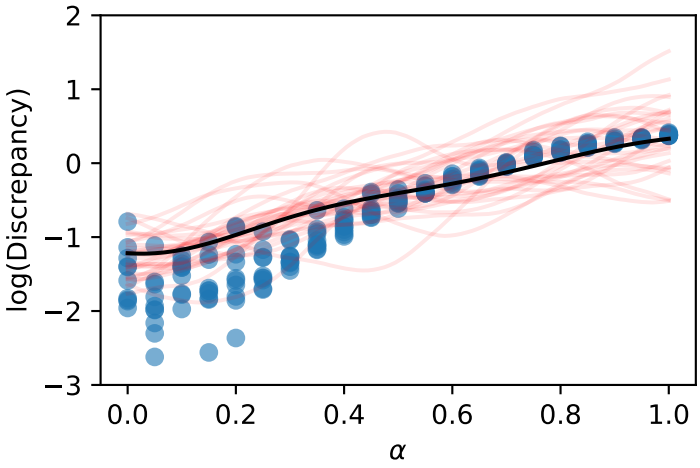

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
β

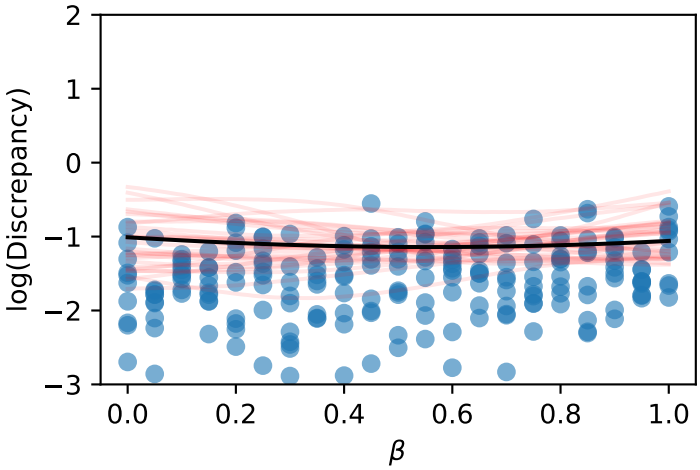

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
γ_L

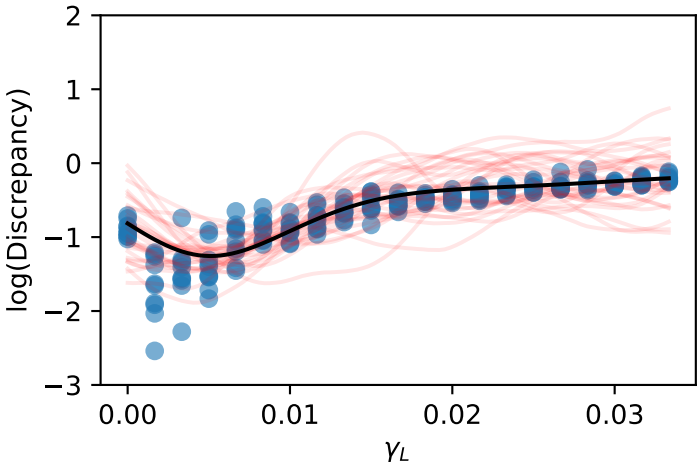

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
0.04
0.05
λ

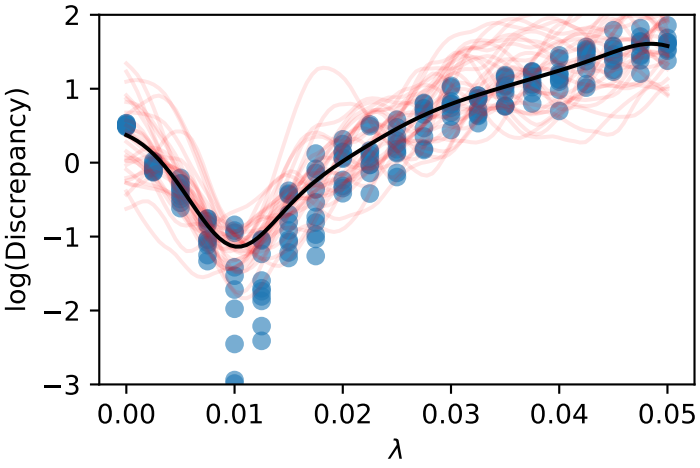

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
0.04
0.05
f

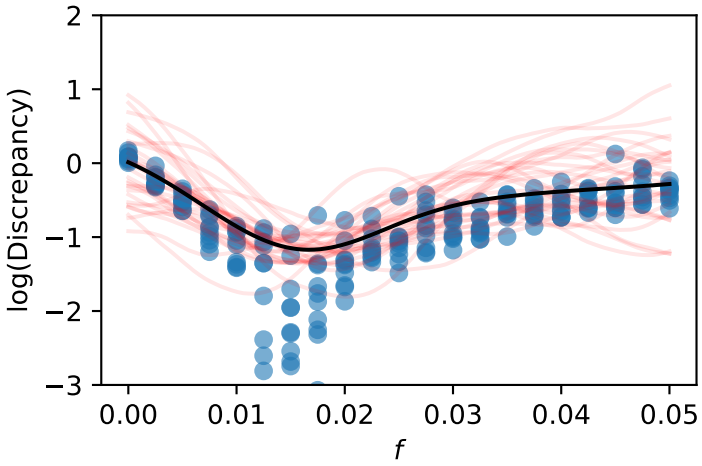

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.02
0.04
0.06
r

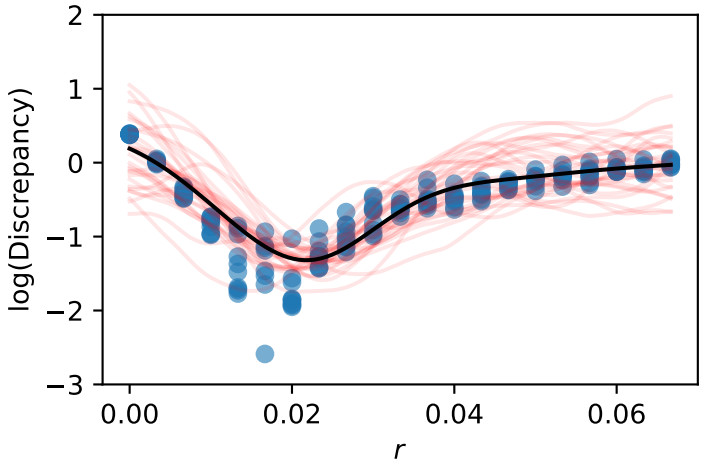

log(Discrepancy)
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
α

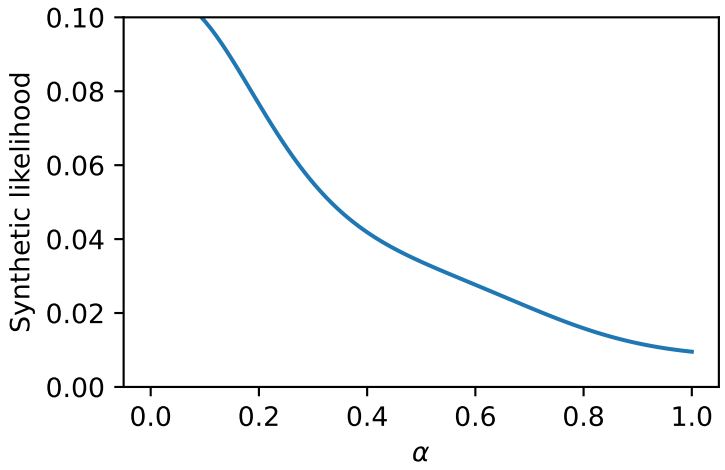

log(Discrepancy)
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
β

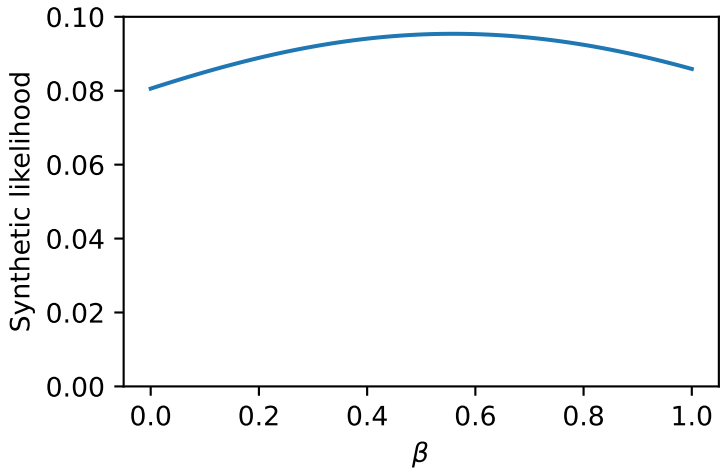

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
γ_L

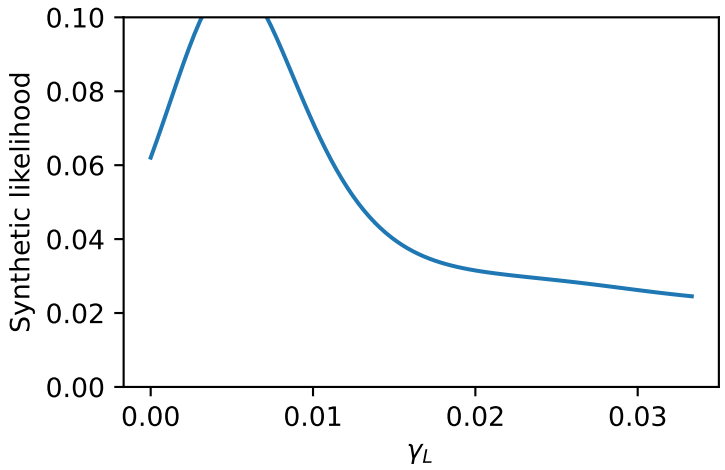

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
λ

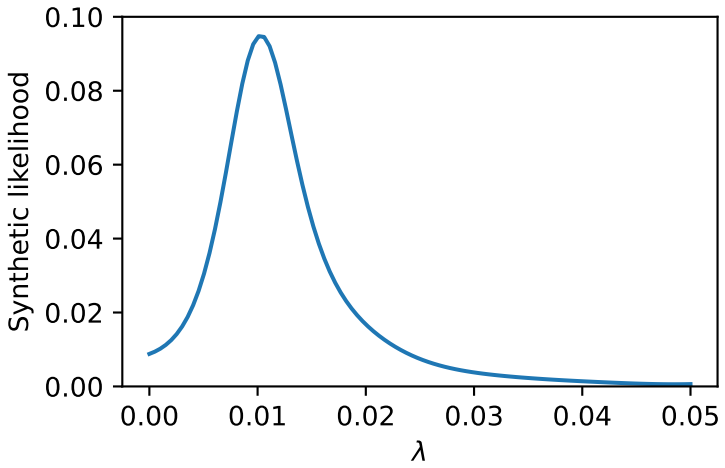

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
f

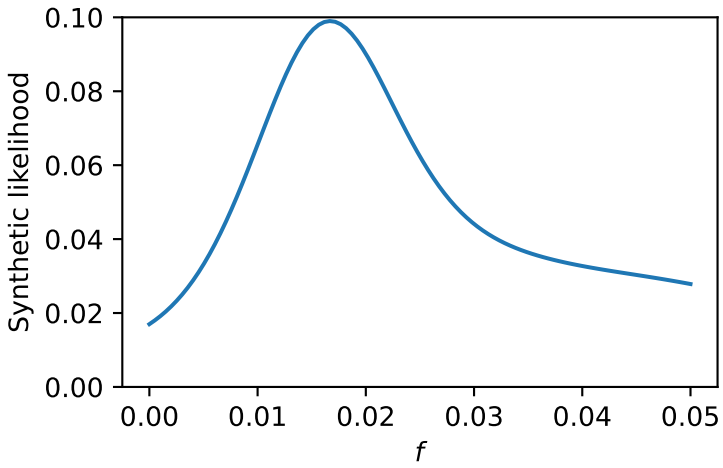

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.02
0.04
0.06
r

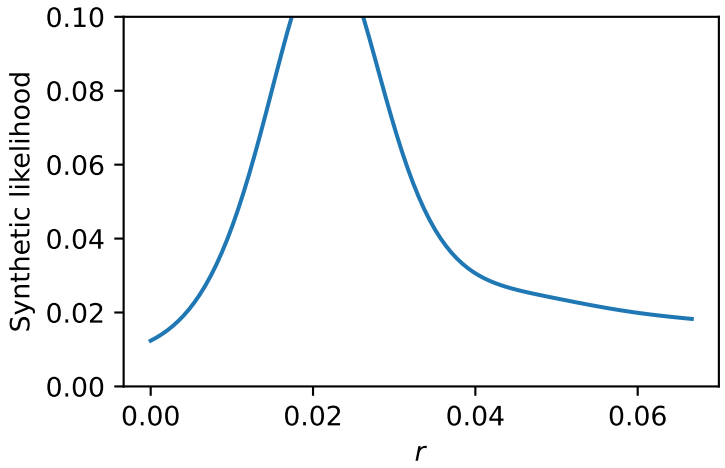

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
α

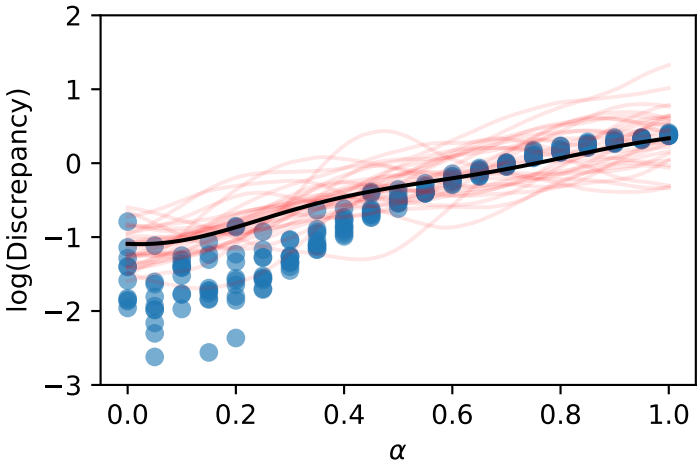

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
β

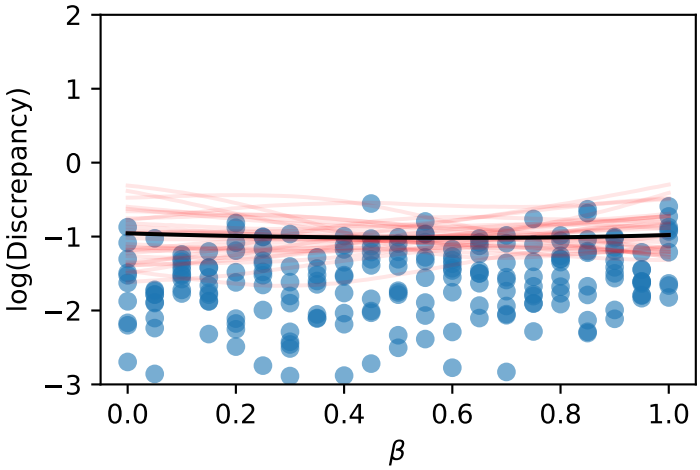

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
γ_L

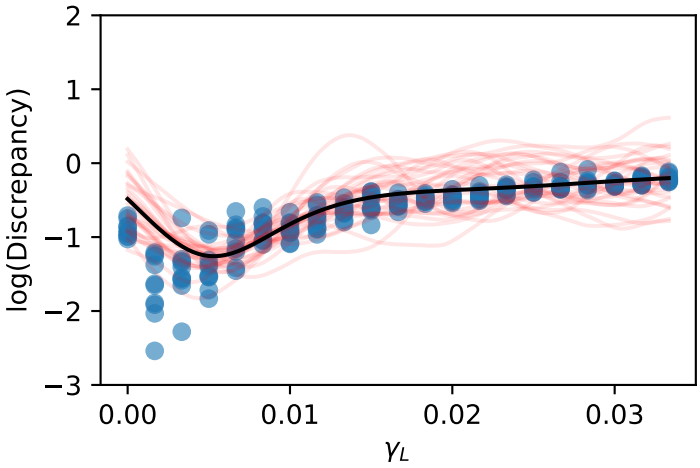

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
0.04
0.05
λ

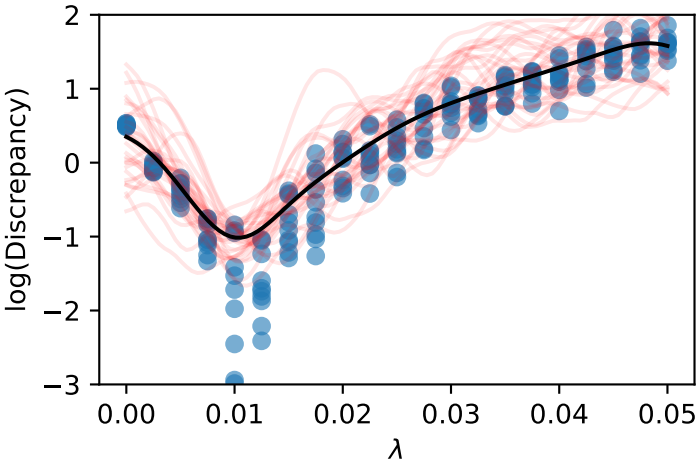

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
0.04
0.05
f

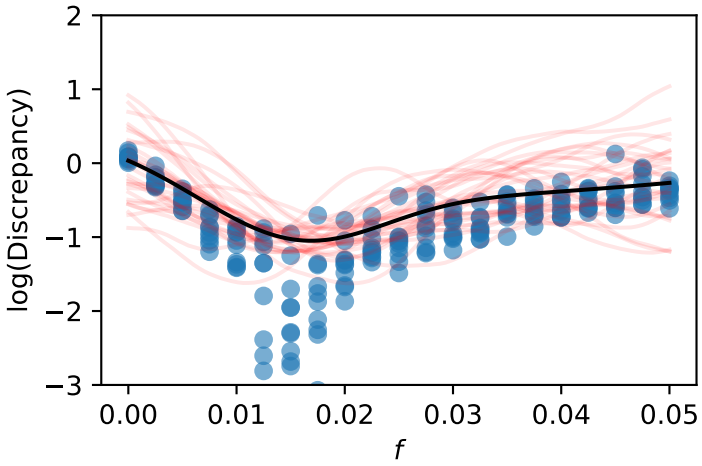

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.02
0.04
0.06
r

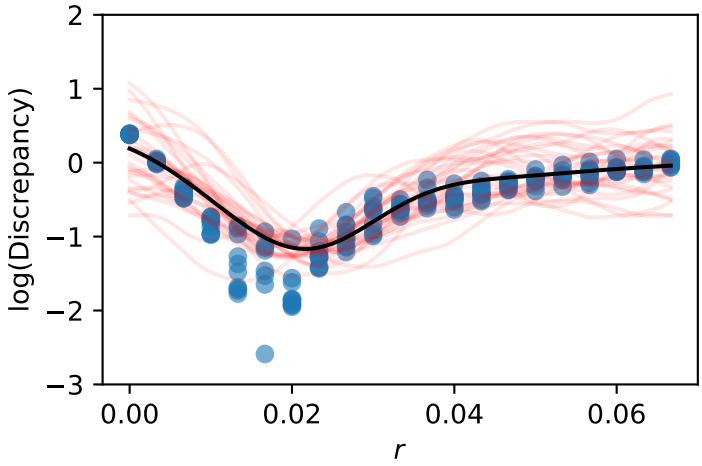

log(Discrepancy)
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
α

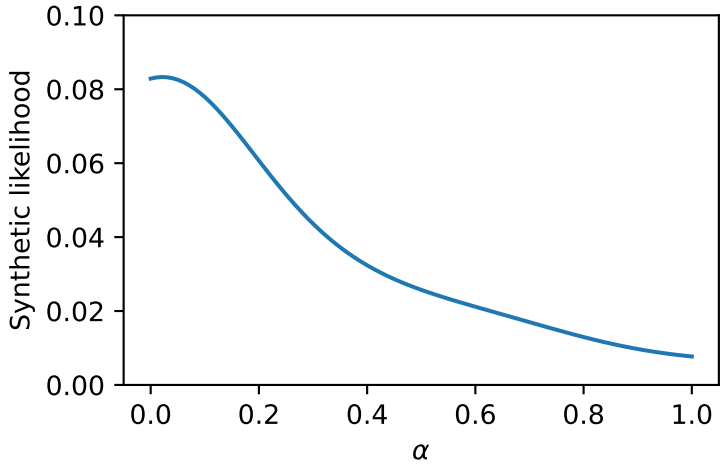

log(Discrepancy)
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
β

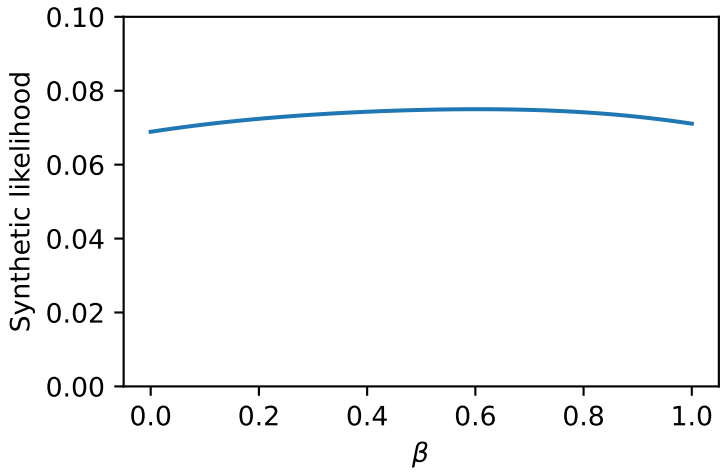

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
γ_L

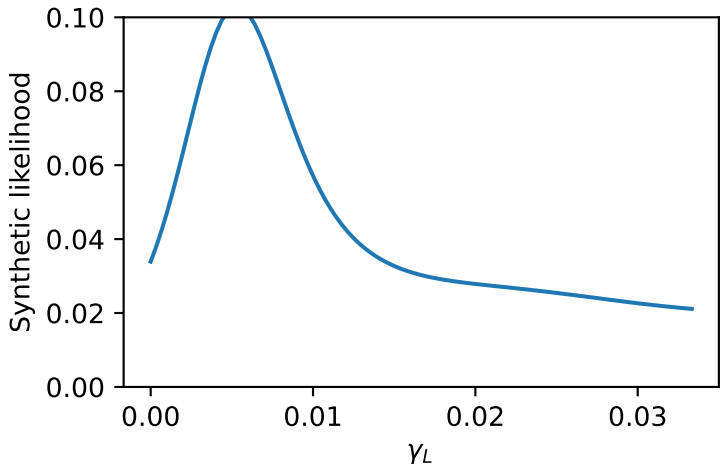

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
λ

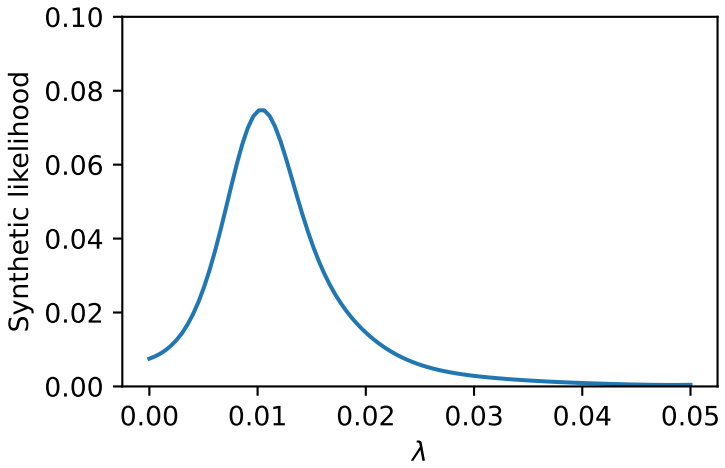

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
f

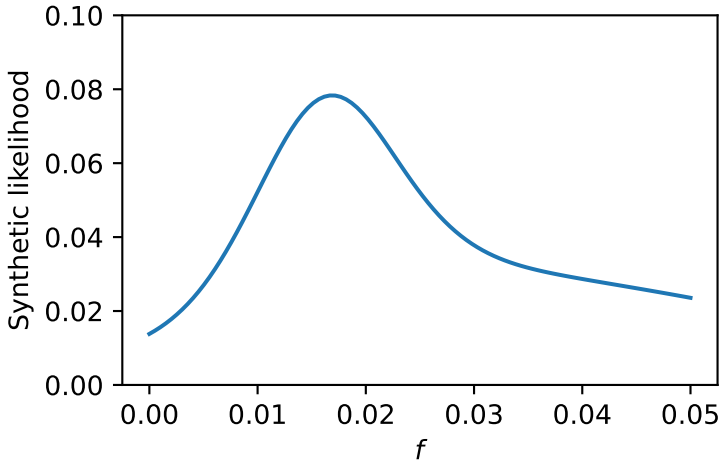

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.02
0.04
0.06
r

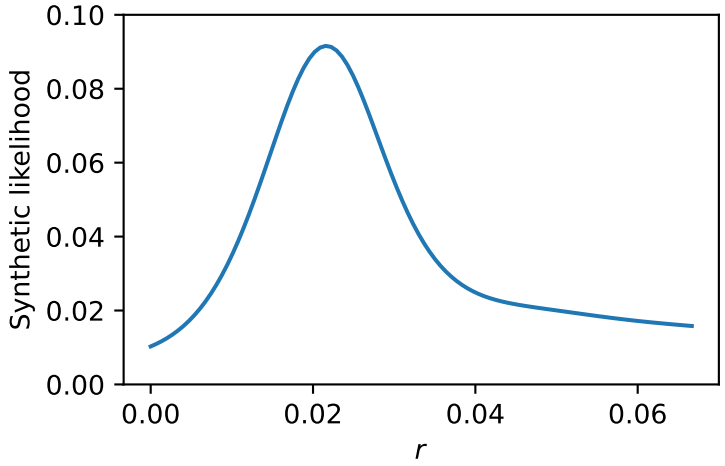

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
α

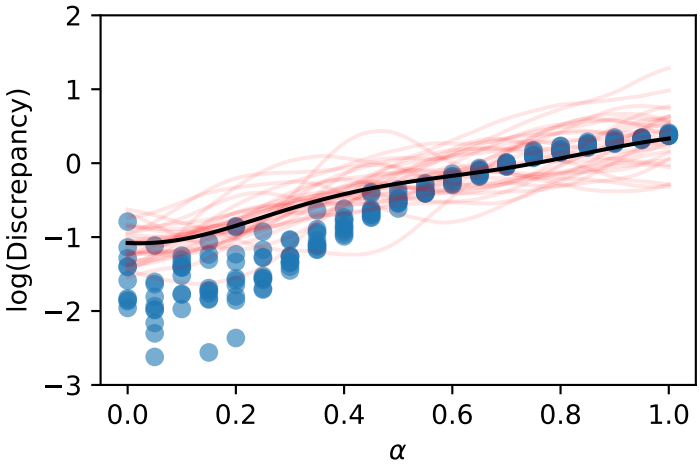

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
β

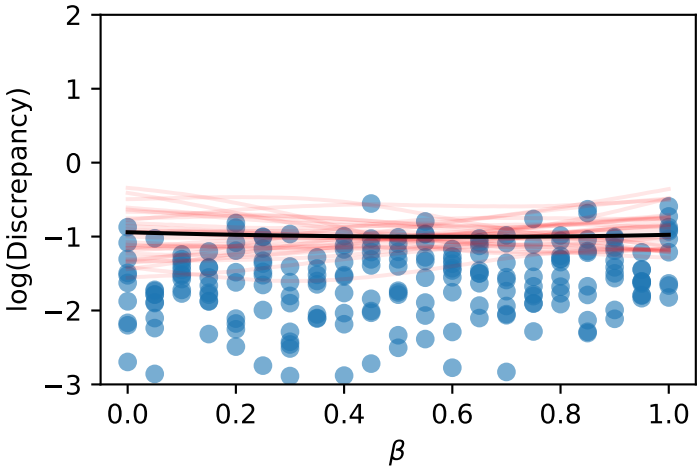

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
γ_L

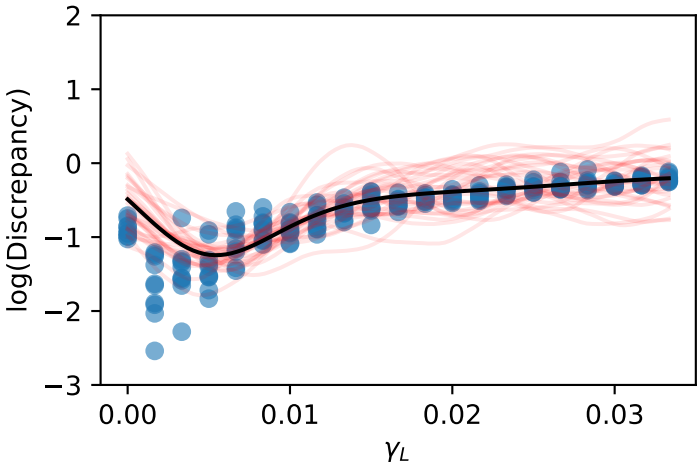

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
0.04
0.05
λ

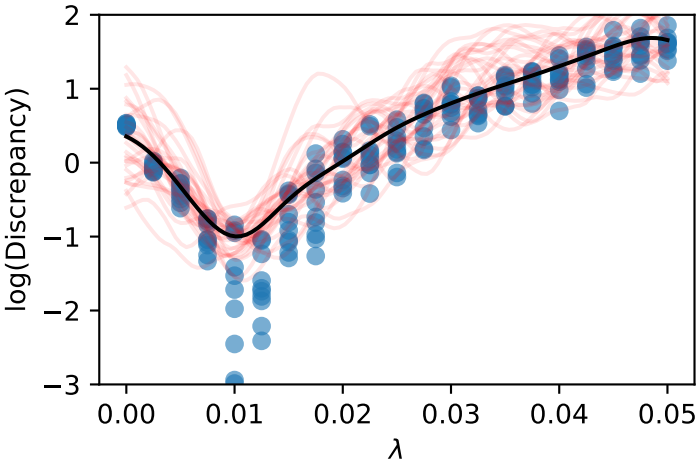

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
0.04
0.05
f

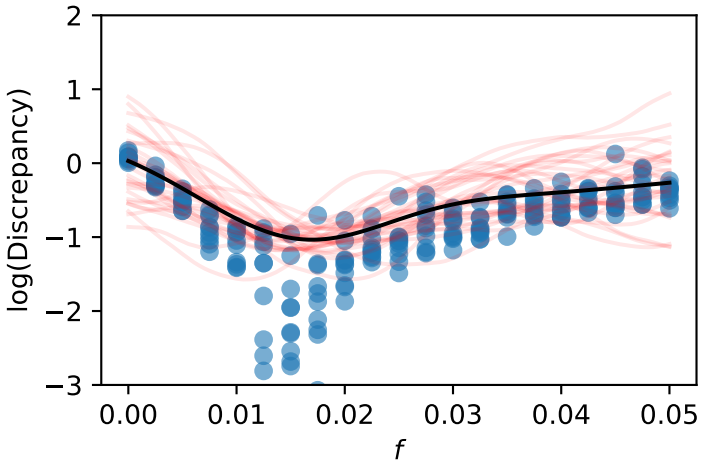

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.02
0.04
0.06
r

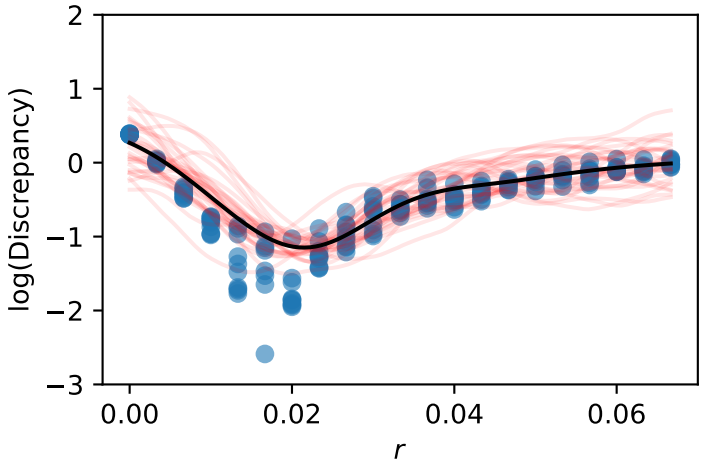

log(Discrepancy)
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
α

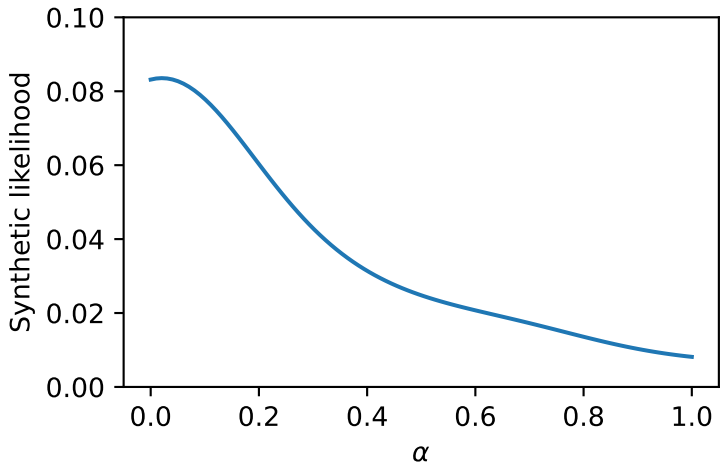

log(Discrepancy)
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
β

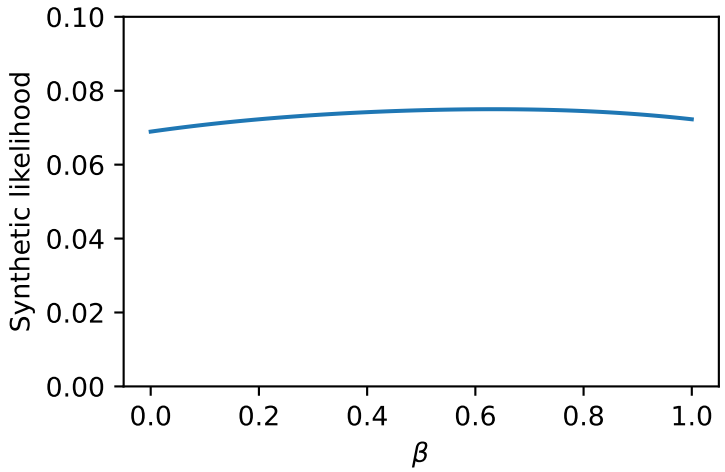

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
γ_L

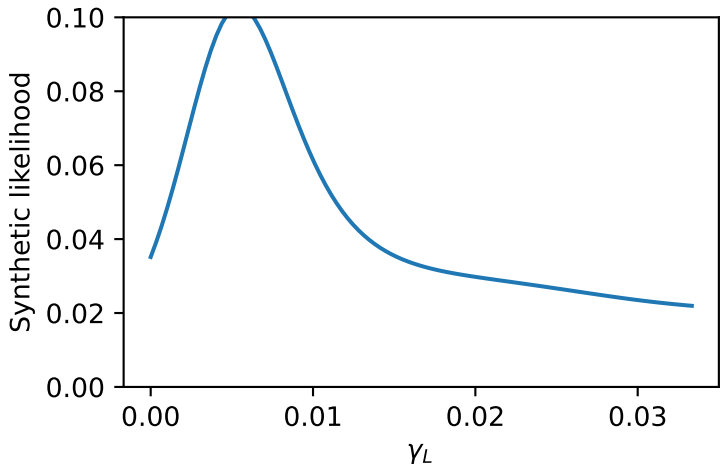

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
λ

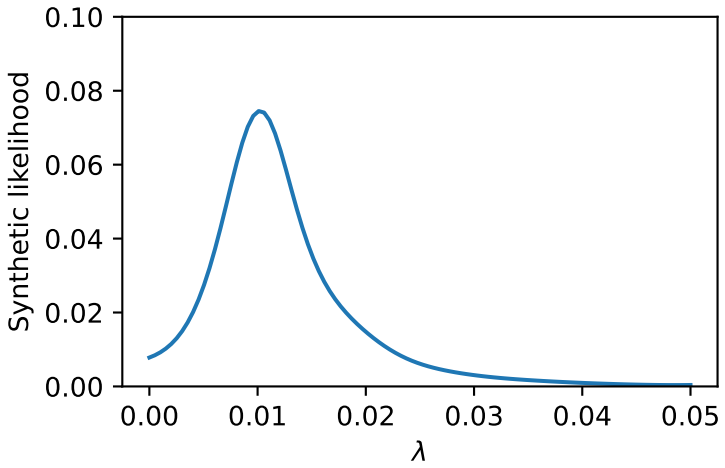

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.01
0.02
0.03
0.04
0.05
f

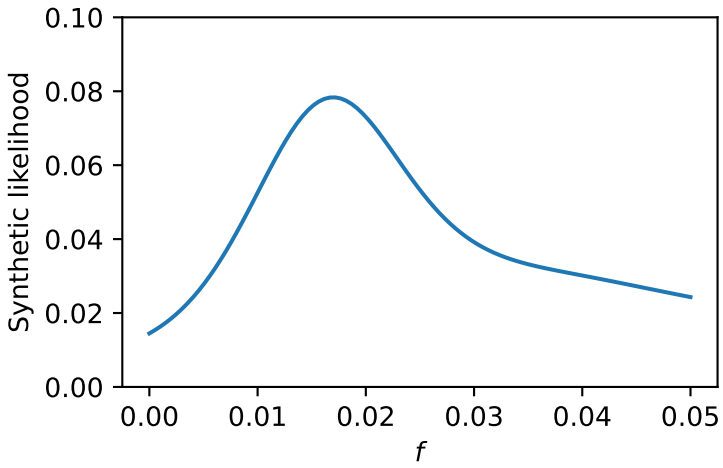

log(Discrepancy)
2
1
0
−1
−2
−3
0.00
0.02
0.04
0.06
r

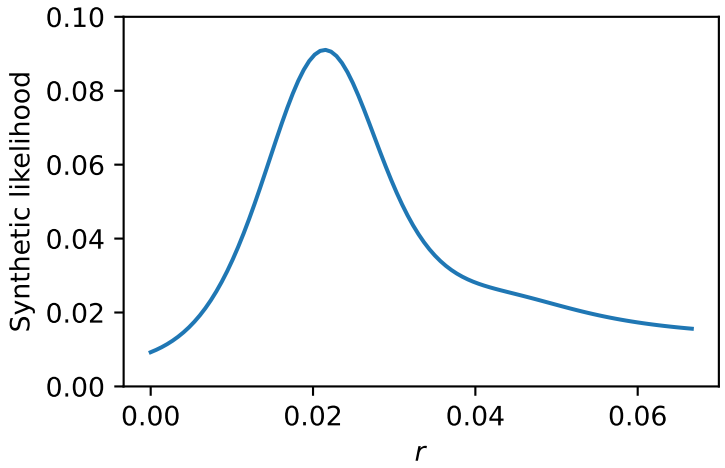

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
α


0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
β


0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
γ_L

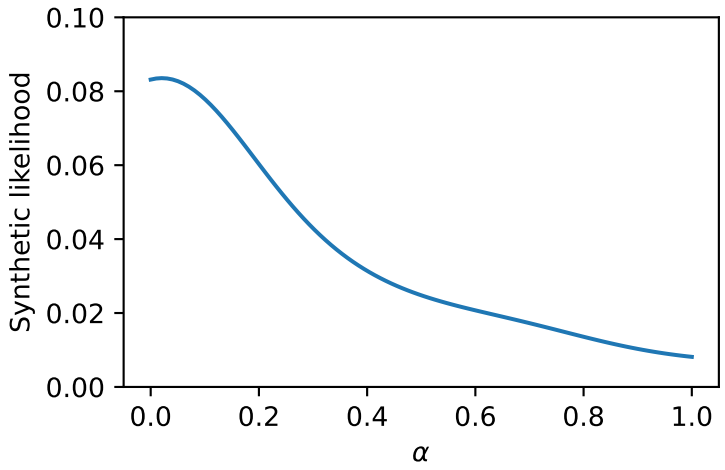

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
0.04
0.05
λ

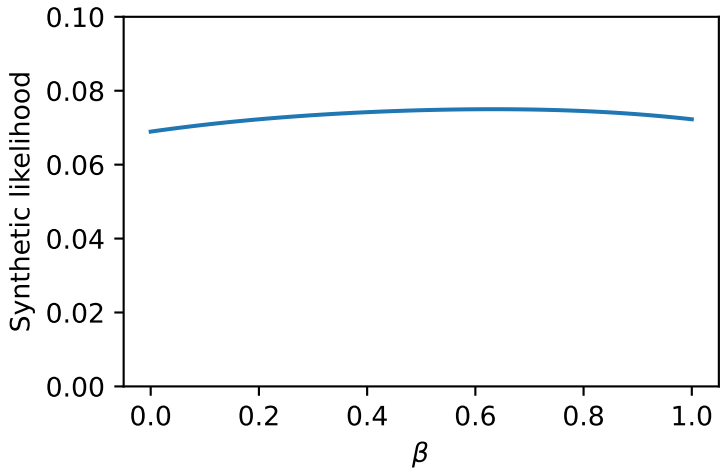


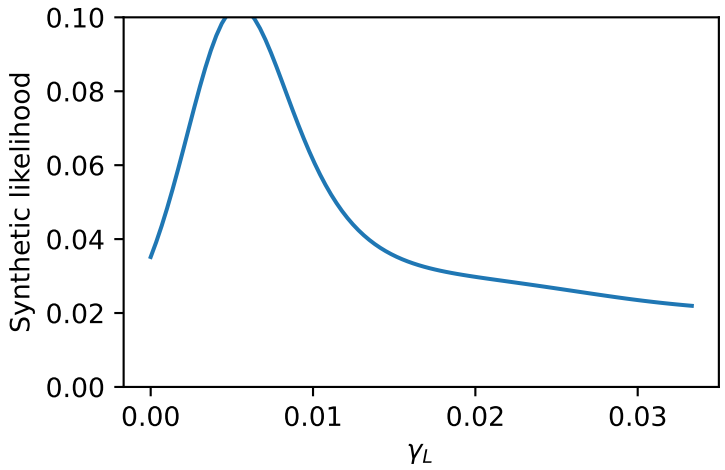


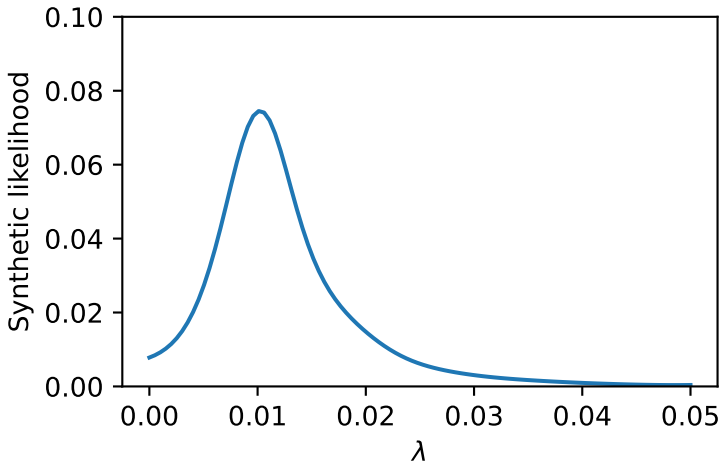

```
epsilon = -3

likelihood_dict = {}
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,
```

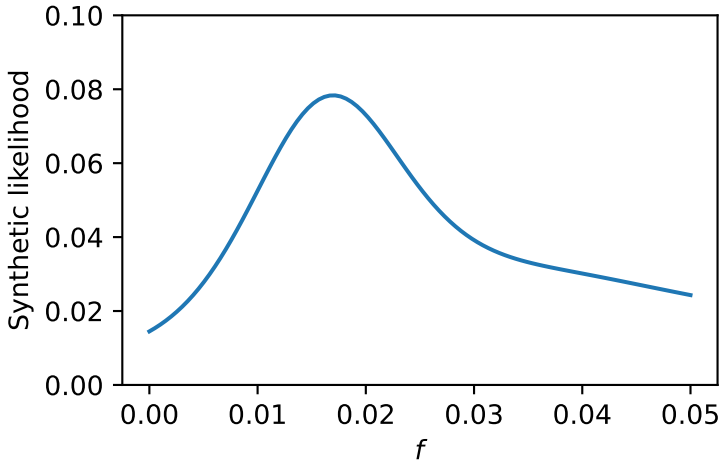
```
    mean_fn=const_mean_fn(),
)

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

mean = champ_GP_reg.mean_fn(indices_for_lik)
likelihood_dict[var + "_slice_means"] = mean
variance = dis_mean_n * observation_noise_variance_champ.numpy()
post_std = np.sqrt(variance)
log_cdf_vals = tfd.Normal(mean, post_std).log_cdf(epsilon)
likelihood_dict[var + "_synth_log_lik"] = log_cdf_vals

plt.figure(figsize=(4, 2.5))
plt.plot(
    slice_indices_dfs_dict[var + "_gp_indices_df"][var].values,
    np.exp(log_cdf_vals),
)
if var in ["f", "r"]:
    plt.xlabel("$" + var + "$")
    # plt.title("Final Synthetic Likelihood for $" + var + "$ Slice")
else:
    plt.xlabel("$\\" + var + "$")
    # plt.title("Final Synthetic Likelihood for $\\" + var + "$ Slice")
plt.ylabel("Synthetic likelihood")
plt.ylim((0, 0.1))
plt.savefig(
    "champagne_GP_images/" + var + "_slice_" + str(t) + "_synth_likelihood.pdf",
    bbox_inches="tight",
)
plt.show()
```

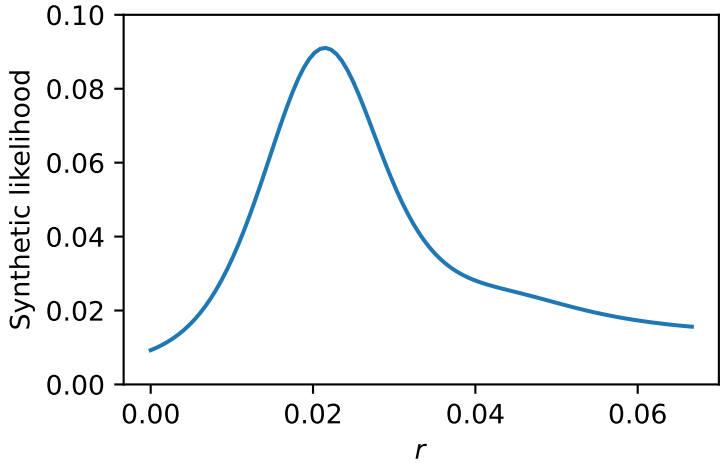

0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
α


0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.0
0.2
0.4
0.6
0.8
1.0
β


0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
γ_L


0.10
0.08
0.06
0.04
0.02
0.00
Synthetic likelihood
0.00
0.01
0.02
0.03
0.04
0.05
λ






```
# 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.    0.001 0.03  0.015 0.036 0.06 ]
[0.998 0.031 0.018 0.023 0.    0.067]
[0.049 0.181 0.006 0.014 0.021 0.024]
[0.    0.001 0.006 0.05  0.018 0.024]
[0.53  0.026 0.03  0.015 0.023 0.022]
[0.049 0.918 0.006 0.022 0.038 0.024]
[0.425 0.002 0.027 0.012 0.046 0.035]
[0.425 0.002 0.027 0.012 0.041 0.035]
[0.425 0.002 0.027 0.012 0.038 0.035]
```

```
objects_to_preserve = [
    index_vals,
    obs_vals,
    champ_samp,
    initial_losses_LOOCV,
    slice_samples_dict,
    slice_discrepencies_dict,
    LHC_indices_df,
    gp_samples_dict,
    likelihood_dict,
]

with open("gp_objs.pkl", "wb") as fp:
    pickle.dump(objects_to_preserve, fp)
    print("dictionary saved successfully to file")
```

```
dictionary saved successfully to file
```