import numpy as np  
from scipy.stats import multivariate\_normal  
  
def prior(x, y, z):  
 # 简化的先验概率，可以根据实际情况调整  
 return 1.0  
  
def likelihood(x, y, z, observed\_position, std):  
 # 简化的似然函数，可以根据实际情况调整  
 mu = observed\_position[:3]  
 cov = np.eye(3) \* std\*\*2  
 position\_distribution = multivariate\_normal(mu, cov)  
 return position\_distribution.pdf([x, y, z])  
  
def bayesian\_update(prior, likelihood, observed\_position, std, x\_range, y\_range, z\_range):  
 posterior = np.zeros((len(x\_range), len(y\_range), len(z\_range)))  
  
 for i, x in enumerate(x\_range):  
 for j, y in enumerate(y\_range):  
 for k, z in enumerate(z\_range):  
 p = prior(x, y, z)  
 l = likelihood(x, y, z, observed\_position, std)  
 posterior[i, j, k] = p \* l  
  
 # 标准化得到后验概率  
 posterior /= np.sum(posterior)  
 # 在 simulate\_bayesian\_inference 函数中的 posterior 生成部分添加  
 # print(posterior.shape)  
  
 return posterior  
  
  
def simulate\_bayesian\_inference(observed\_position, std, x\_range, y\_range, z\_range):  
 posterior = bayesian\_update(prior, likelihood, observed\_position, std, x\_range, y\_range, z\_range)  
 # 在 simulate\_bayesian\_inference 函数中的 posterior 生成部分添加  
 # print(x\_range)  
 # print(y\_range)  
 # print(z\_range)  
 return posterior