结构说明

1.core核心逻辑

1.1 algorithms

1.1.1 genetic.py遗传算法部分

* 输入：配置文件config；初始环境env类；目标分布gp类；障碍物obs类
* 输出：搜索路径path（数据类型：array）；训练好的环境trained\_env类（也可不用输出）；可能要用数组记录累计探测概率的变化
* 内容：算法主体；生成路径的函数

1.1.2 greedy.py贪婪算法部分

* 输入：配置文件config；初始环境env类；目标分布gp类；障碍物obs类
* 输出：搜索路径path（数据类型：array）；训练好的环境trained\_env类（也可不用输出）；可能要用数组记录累计探测概率的变化
* 内容：算法主体；生成路径的函数

1.1.3 rl.py强化学习算法部分

* 输入：配置文件config；初始环境env类；目标分布gp类；障碍物obs类
* 输出：搜索路径path（二维坐标列表，数据类型：array）；训练好的环境trained\_env类；可能要用数组记录累计探测概率的变化
* 内容：算法主体；生成路径的函数

1.2 environment环境生成

1.2.1 poc目标概率分布生成

1.2.1.1 generate\_poc.py

* 内容：定义GeneratePOC类；参数包含所有与目标概率分布相关的设置，如poc\_type（目标分布概率的类型，用config["poc\_type"]读取）、poc（数据类型ndarray）、gauss\_center（正态分布中心坐标，数据类型array）……；通过读取poc\_type调用不同的路径生成函数。除了gaussian、0-1、uniform这三种基础分布，新增dynamic\_gaussian分布。

1.2.1.2 gaussian.py

* 内容：①定义函数initialize\_gauss\_poc，生成一个length\*width的正态分布概率分布，用数组存储（静态目标）。  
  ②定义函数update\_dynamic\_gaussian\_poc，随时间变化，给出目标点的运动轨迹，接下来调用initialize\_dynamic\_gauss\_poc函数生成当前时刻下的poc分布，如果是迭代训练的话可以在每个时间步调用函数initialize\_poc来更新全局的poc地图；定义函数initialize\_dynamic\_gauss\_poc，根据传入的目标点位置，更新当前时刻下以目标点为中心的poc概率分布，嵌套在update\_dynamic\_gaussian\_poc函数中；定义函数get\_current\_target，用于计算当前时刻下目标点的位置，核心算法需要目标位置时调用即可（动态目标）。

1.2.1.3 zero\_one.py

* 内容：定义函数initialize\_01\_poc，生成一个length\*width的0-1分布概率分布，用数组存储

1.2.1.4 unform.py

* 内容：定义函数initialize\_uniform\_poc，生成一个length\*width的均匀分布概率分布，用数组存储

1.2.2 grid.py网格生成

* 内容：定义GridEnvironment类；参数包括网格大小length和width（数据类型int）、动作空间（8个坐标，用array存储）、current\_time当前时间、total\_time总时长、~~搜索起点（坐标，用array存储）~~。定义函数update\_time，在算法的每个时间步调用，用于计算当前时间步对应的时刻（是否调用这个函数可自行决定，直接用当前的时间步做后续处理也可以）。

1.2.3obstacles.py障碍物生成

* 内容：定义ObstacleManager类；参数包括障碍物地图obstacles（用array存储）、障碍物类型obstacle\_type（用config["obstacle\_type"]读取）；定义生成固定障碍物函数initialize\_fixed\_obstacles、生成随机障碍物函数initialize\_random\_obstacles等

1.3 utils

1.3.1 visualize.py可视化函数

* 输入：训练好的环境trained\_env类（包括cdp\_list累计探测概率记录表和reward\_list训练奖励记录表）、障碍物obs类、目标分布概率gp类、搜索路径path（一连串二维坐标，数据类型：array）
* 输出：目标概率分布热力图、累计探测概率变化折线图（考虑中，仍需讨论）、~~搜索成功概率热力图、~~搜索路径、各关键点
* 内容：定义Visualize可视化类；定义visualize可视化函数；定义函数visualize\_cdp\_and\_reward，用于画出累计探测概率（和训练奖励，如果没有可以全为0）的变化折线图。

2. data数据存储

3. ui人机交互

3.1 config\_parser.py

* 输入：无
* 输出：无
* 内容：各类参数，包括环境初始化、算法所需

**3.2 gui**

4.main.py

主函数，逻辑为：实例化各种class类->初始化环境、障碍物->设置目标概率分布->选择算法并执行->得到路径并存储在path->将path传入可视化函数模块，生成图像