# **地下综合管廊智能巡检与路径规划系统 V1.0 软件说明书**

## **1. 摘要**

本软件面向城市地下综合管廊巡检场景，研发了一套集自主定位仿真、环境感知、风险场建模、视觉缺陷检测、风险感知路径规划与动态重规划为一体的智能巡检系统。系统通过网格化地图、风险源高斯叠加模型形成连续风险场，将风险值引入路径代价函数，提出风险场驱动的多目标路径规划（Risk-A\*），并在环境突变时触发在线重规划。系统提供可视化演示界面，支持伪-YOLO 与真实 YOLO 接口替换，可用于高校教学、科研验证与智慧城市运维方案的原型演示。

## **2. 软件开发目的与应用范围**

1. 用以验证地下管廊巡检机器人在无 GPS 环境下的路径规划与风险避让算法。

2. 为视觉缺陷检测（裂缝/积水/烟雾）提供仿真评估与接口替换方案。

3. 为智慧城市地下运维方案提供原型演示与教学平台。

## **3. 功能列表**

·三维/二维可视化：显示风险热力图、机器人轨迹、规划路径、检测框。

·地图管理：随机生成管廊结构并可在线添加/移除障碍。

·风险场计算：支持静态与动态风险源，输出每栅格风险值。

·SLAM 模拟：提供估计位姿与真实位姿（带噪声）。

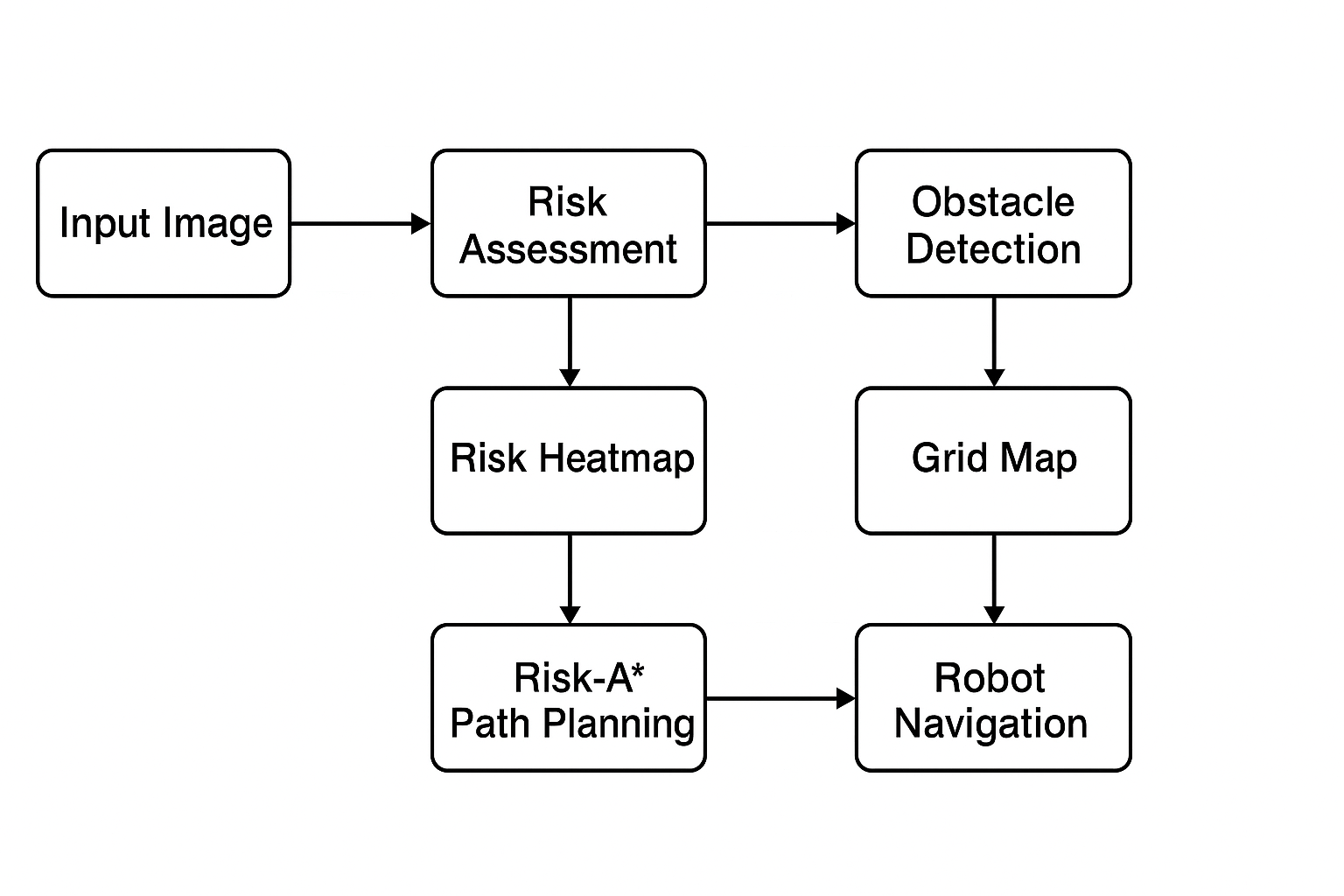
·视觉检测：伪-YOLO 生成 bbox/score，可替换为真实 YOLOv5/YOLOv8。

·风险感知路径规划：Risk-A\* 算法（代价含长度/风险/能耗）。

·仿真控制：步进仿真、触发重规划、结果报告导出。

## **4. 系统架构与流程**

系统采用“感知—决策—执行—展示”四层架构：感知层负责环境数据与相机帧；感知融合层计算风险场并输出视觉检测结果；决策层进行风险评估与路径规划（Risk-A\*）并在必要时触发重规划；执行层模拟机器人运动与 SLAM；展示层进行实时可视化与日志保存。



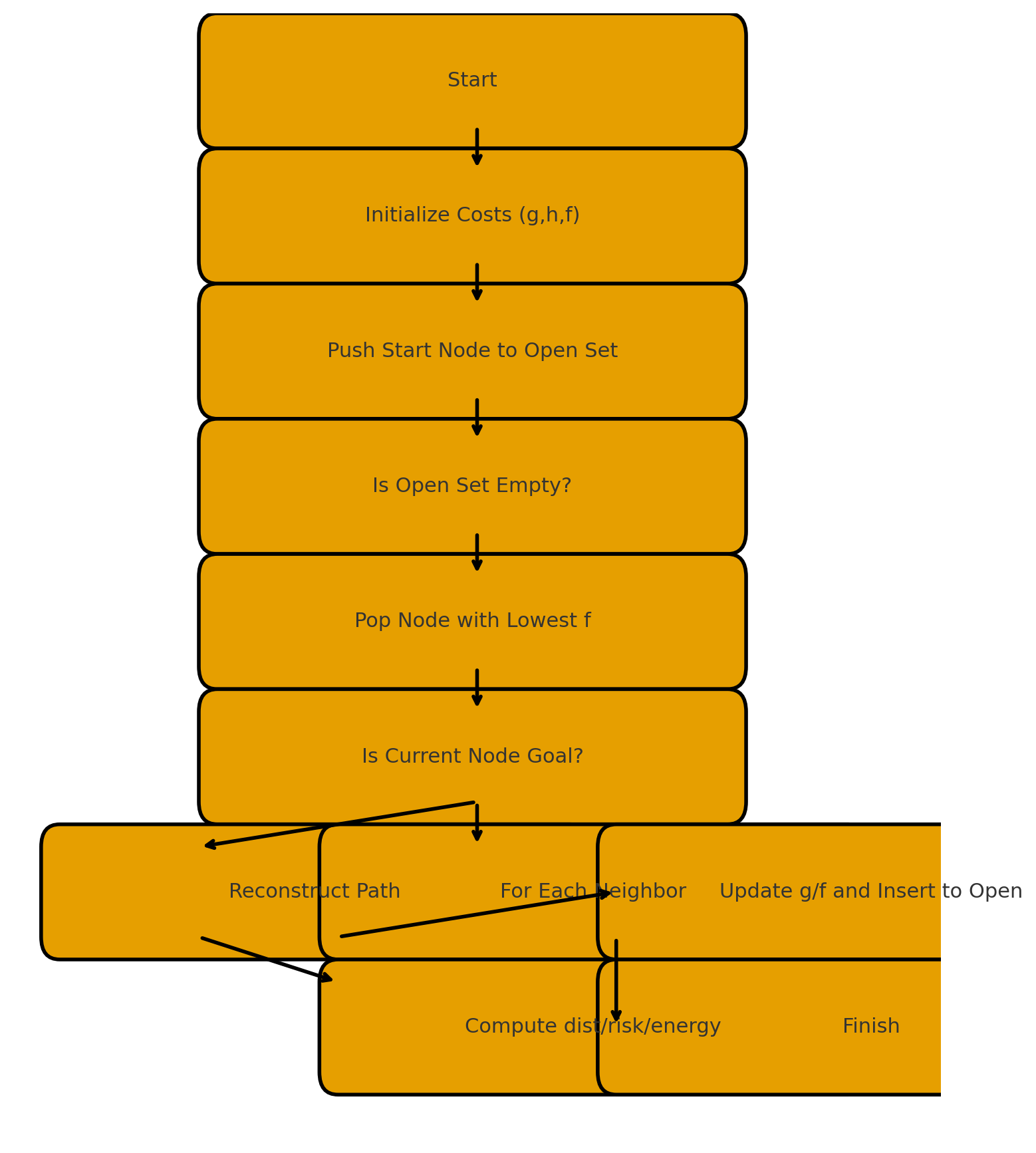
流程说明：

本系统首先对输入图像进行风险评估，提取潜在风险源信息，并根据风险因子的间分布生成风险热力图。同时，系统通过视觉或传感器检测对环境进行障碍物识别并构建栅格地图。随后，Risk-A\* 路径规划模块综合考虑路径代价与风险权重，生成避开高风险区的安全轨迹，最终由机器人导航模块执行运动控制，实现智能巡检。

## **5. 主要算法说明**

1.风险场模型：

2.风险感知代价函数：

**流程说明：**  
 Risk-A\* 算法基于传统 A\*，在启发式搜索过程中加入风险场约束。算法首先初始化各节点的代价（g、h、f），并将起点加入 Open 集合。在循环中，从 Open 集合中取出 f 值最小的节点作为当前节点；若当前节点为目标点，则执行路径重建。若不是目标，则遍历其相邻节点，根据路径距离、风险值和能耗计算代价，并更新 g/f 值后加入 Open 集合。循环持续进行直至找到最优路径或搜索失败。

算法 1：Risk-A\* 路径规划伪代码

（本算法为本软件核心创新点之一）

Algorithm RiskAStar(start, goal, Grid, RiskField)

open\_set ← priority queue ordered by f = g + h

closed\_set ← empty set

g\_score[start] ← 0

f\_score[start] ← Heuristic(start, goal)

open\_set.insert(start, f\_score[start])

while open\_set not empty:

current ← open\_set.pop\_min()

if current == goal:

return ReconstructPath(current)

closed\_set.add(current)

for each neighbor in Grid.Neighbors(current):

if neighbor in closed\_set:

Continue

dist\_cost ← Distance(current, neighbor)

risk\_cost ← RiskField(neighbor)

energy\_cost ← EnergyEstimate(neighbor)

# 综合代价函数

total\_cost ← α·dist\_cost + β·risk\_cost + γ·energy\_cost

tentative\_g ← g\_score[current] + total\_cost

if neighbor not in open\_set or tentative\_g < g\_score[neighbor]:

parent[neighbor] ← current

g\_score[neighbor] ← tentative\_g

f\_score[neighbor] ← tentative\_g + Heuristic(neighbor, goal)

open\_set.insert\_or\_update(neighbor, f\_score[neighbor])

return Failure

## **6. 数据结构与接口**

### 6.1 数据结构说明

#### （1）Node 结构（用于 Risk-A\* 搜索）

用于表示路径规划中的单个节点，包含以下字段：

**x, y**：节点所在的网格坐标

**g\_cost**：从起点到当前节点的累计代价

**h\_cost**：当前节点到目标节点的启发式估价

**f\_cost**：总代价，f = g + h

**parent**：指向父节点，用于路径回溯

#### （2）RiskField 结构（风险场数据表示）

用于存储和管理二维风险场：

**risk\_map**：二维浮点矩阵（size: M × N），表示风险值

**sources**：风险源列表，每个风险源形如：  
{ position, intensity, radius, volatile\_factor }

**resolution**：网格分辨率（米/格）

#### (3) GridMap 结构（环境栅格地图）

**grid**：二维整型矩阵（0=空闲；100=障碍）

**width, height**：地图尺寸

**origin**：坐标原点

**resolution**：网格大小

#### （4）Detection 结构（YOLO 输出）

**bbox**：边界框 (xmin, ymin, xmax, ymax)

**score**：置信度（0–1）

**class\_id**：类别 ID（0=裂缝，1=渗漏，2=异物 等）

**label**：可读标签（crack / leak / foreign-object）

### 6.2接口说明

RiskField.add\_source(pos, intensity, radius, volatile)

RiskField.compute() → 返回二维数组风险矩阵

RiskAStar.plan(start, goal) → 返回路径点列表

PseudoYOLO.detect(cell, fov\_cells) → 返回检测框列表

## **7. 用户操作与运行说明**

### 7.1 运行环境要求

本软件基于 Python 语言开发，运行环境如下：

操作系统：Windows 10 / Windows 11 / Linux（Ubuntu 20.04 及以上）

语言环境：Python 3.8 或更高版本

依赖库：

·numpy（数据计算）

·opencv-python（图像读取与处理）

·matplotlib（可视化）

·torch（YOLO 模型推理，若使用真实模型）

硬件要求：

·CPU：Intel i5 或以上

·内存：8GB 或以上  
-（可选）NVIDIA GPU 用于加速检测模型

### 7.2 安装与运行步骤

1.安装 Python 3.8+，并安装依赖：

pip install numpy matplotlib opencv-python torch

2.下载文件 underground\_inspection\_robot\_yolo.py。

3.运行程序：

python underground\_inspection\_robot\_yolo.py

4.程序运行完成后，会自动生成检测与规划结果报告文件：  
simulation\_report\_yolo.txt。

## **8. 软件运行环境与依赖**

操作系统：Windows 11/ Ubuntu 20.04

开发语言：Python 3.8+

开发工具：PyCharm / VSCode / Jupyter Notebook

依赖组件：

·numpy —— 数值计算

·atplotlib —— 可视化绘制

·opencv-python —— 图像读取与预处理

·ultralytics（YOLOv8 推理）或 torch（YOLOv5 推理）

·（可选）CUDA + cuDNN —— GPU 推理加速

## **9.源代码清单**

本软件包含一个主程序文件：

1）underground\_inspection\_robot.py

系统的核心程序，集成以下功能模块：

·**GridMap：**地下管廊环境建模

·**RiskField / RiskSource：**二维高斯风险场模拟

·**RiskAStar**：基于风险的 A\* 路径规划

·**SLAMMock**：简化 SLAM 位置估计

·**VisionSim：**裂缝 / 渗漏 / 异物的检测模拟

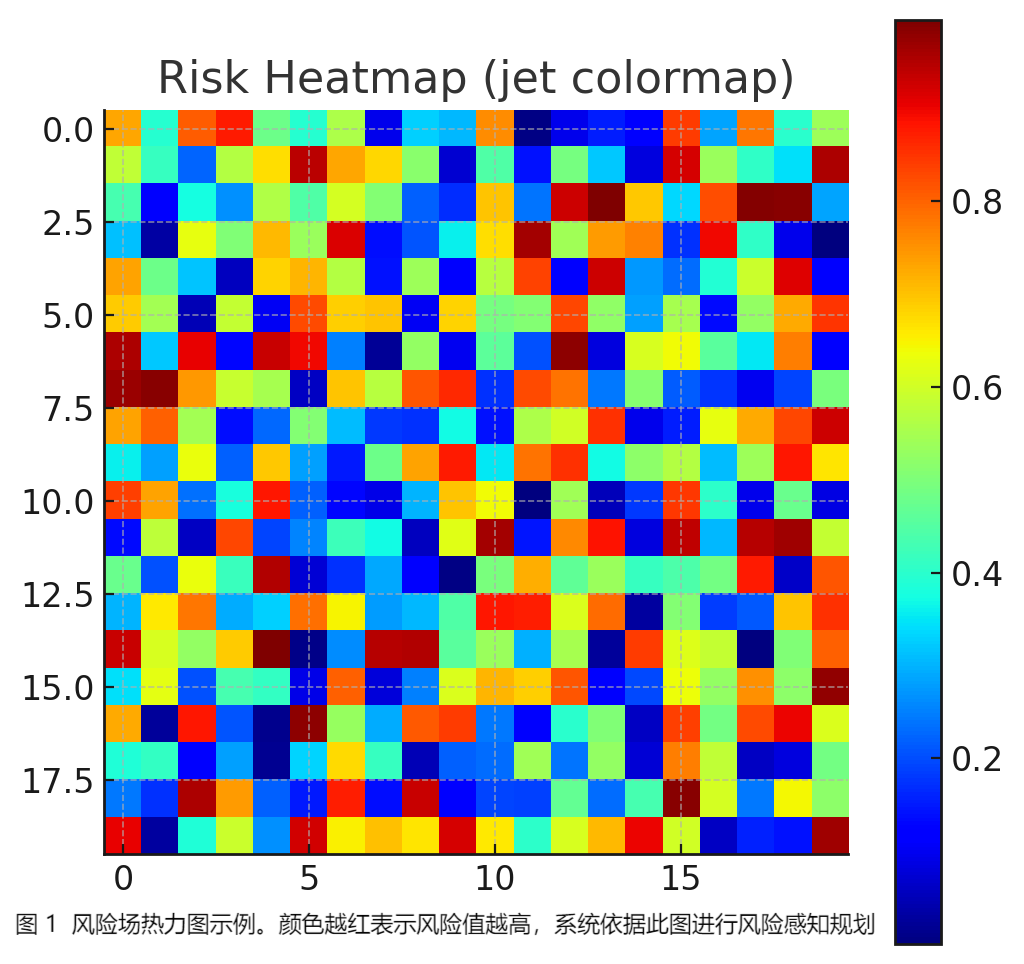
·**Robot：**机器人运动、传感器、日志管理

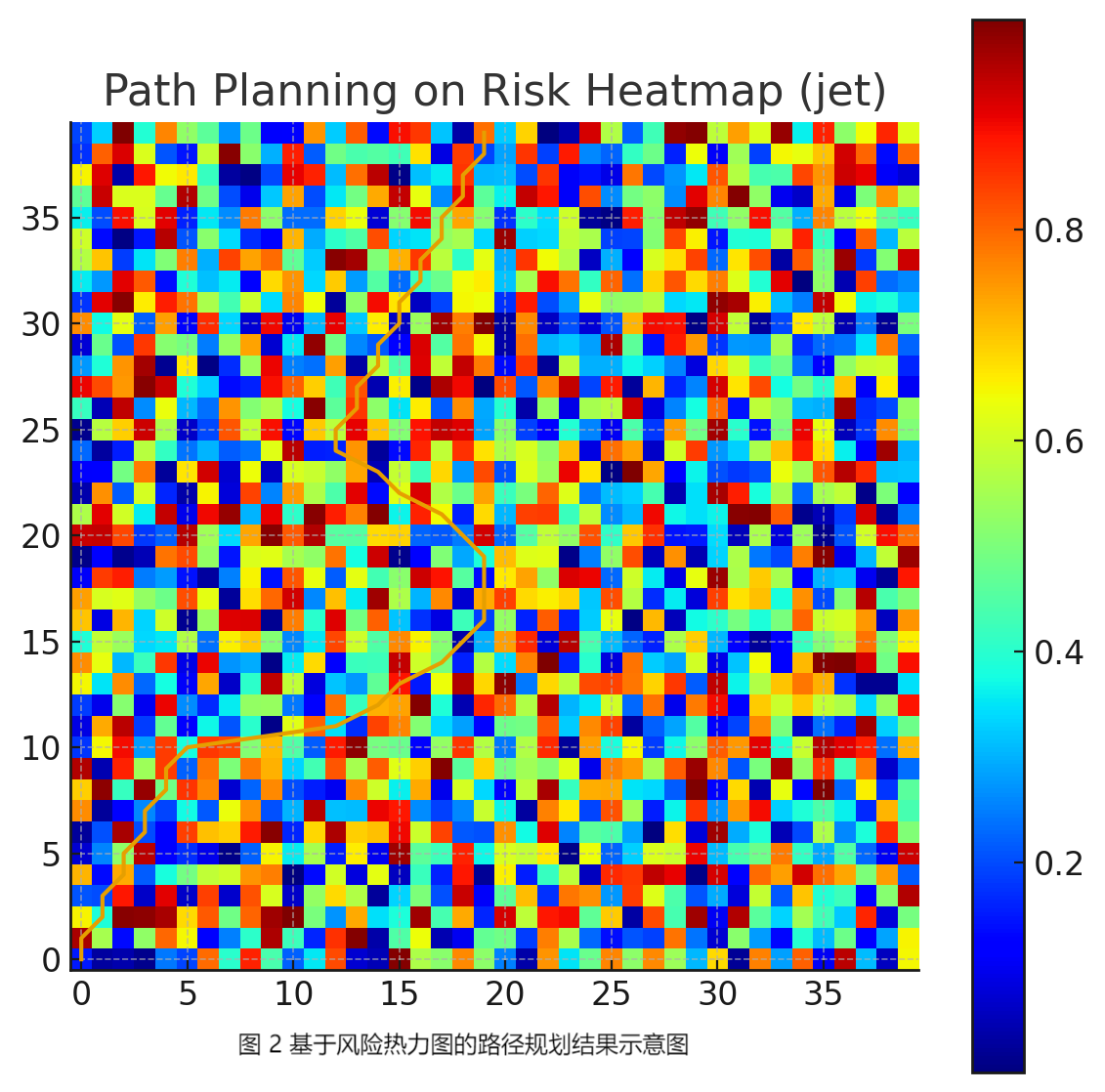
·**Simulator：**调度机器人执行任务

·**Visualizer：Matplotlib：**动态风险场 + 路径可视化

该文件是完整的软件运行入口，执行方式：

python underground\_inspection\_robot.py

10.测试截图与报告



## **11. 版权声明**

本软件由河北工业大学 地下管廊团队开发，具有完整知识产权。未经授权，不得商用或传播。