



# 窥见未来

——AIoT技术前沿与Home Assistant系统实战

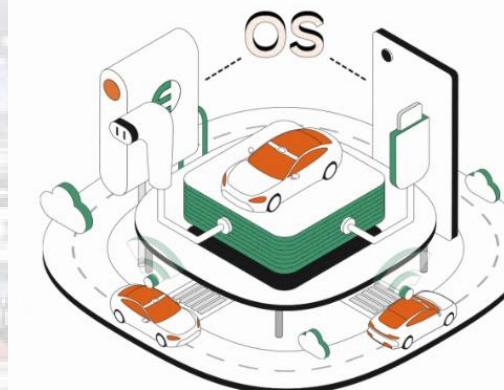


2025年4月





发展智能网联  
新能源汽车产业生态



加快推进人工智能  
终端产业高质量发展



优化新能源  
汽车号牌设计

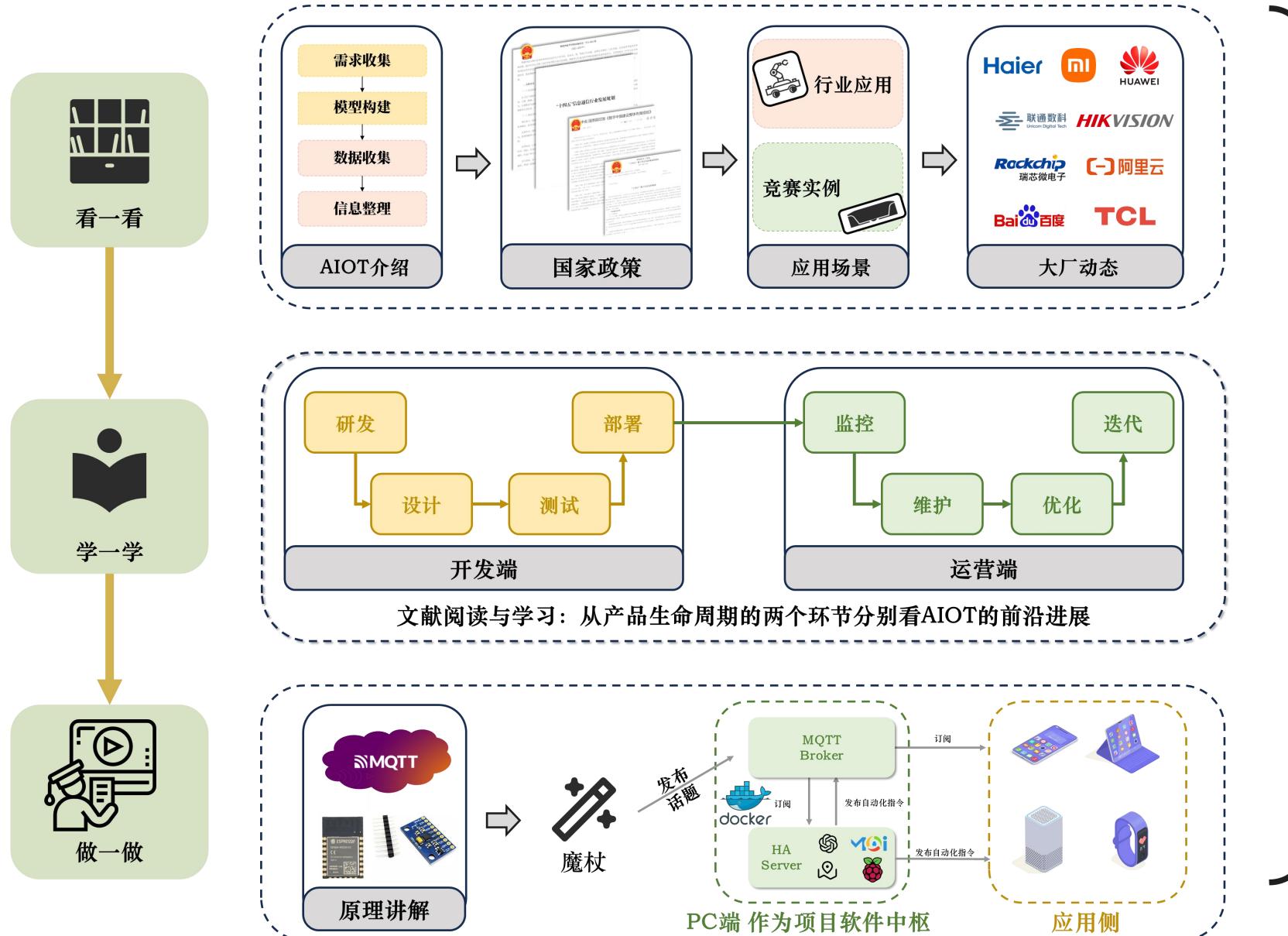


加快推进  
自动驾驶量产



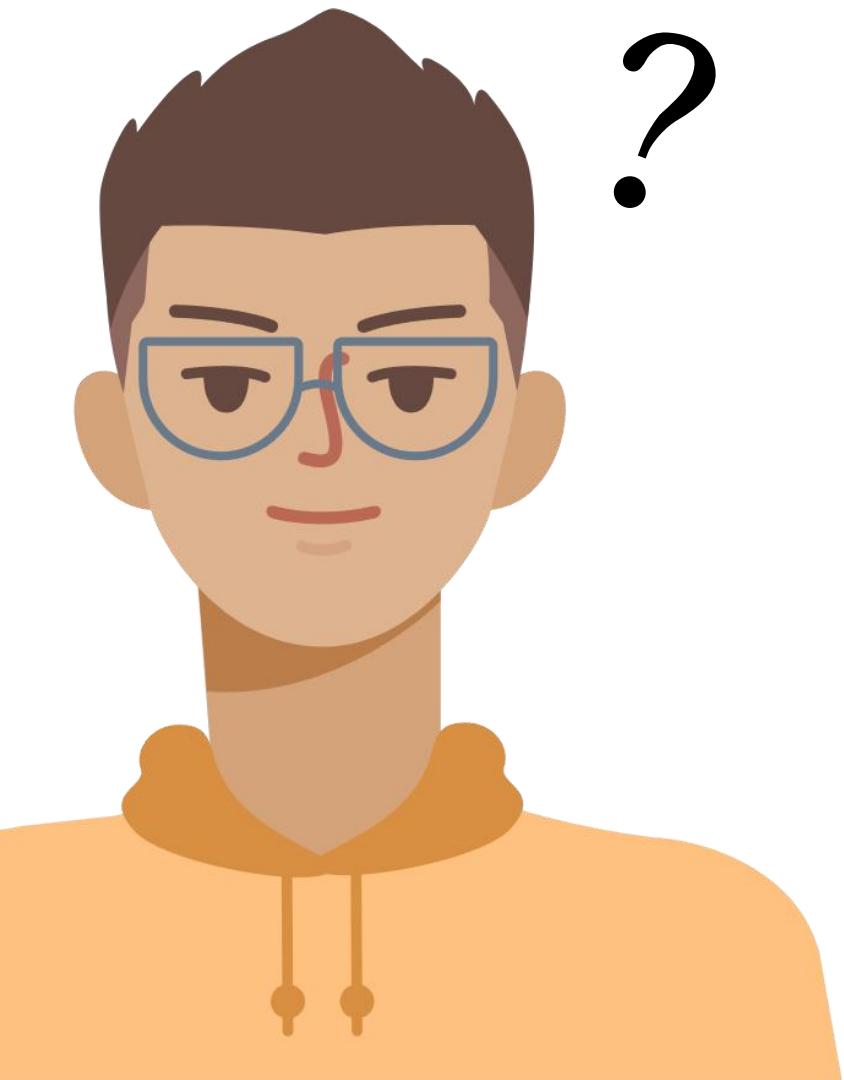
加强“AI换脸拟声”  
违法侵权重灾区治理





# Part 1 看一看

基础介绍、国家政策、应用场景、大厂动向



什么是AIOT?



# AIoT 有广泛应用场景，覆盖千行百业

**AI+**

探索人工智能自身能力



RNN-T模型的准确率达到96.8%，  
超越了人类 (94.17%)

BERT模型的准确率达到90.9%，  
超过了人类 (82%)

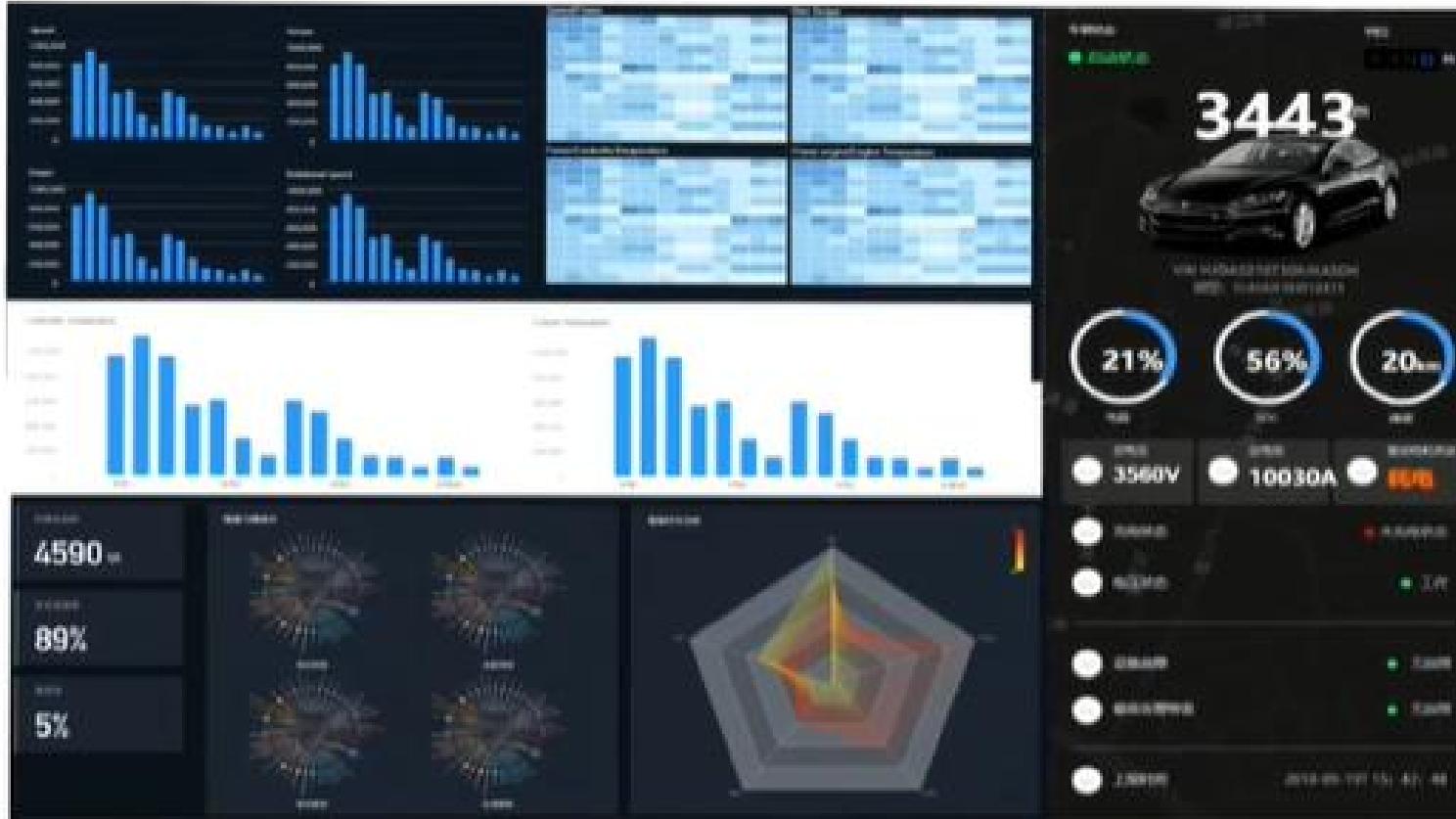
EfficientNet模型的Top-5准确率达到98.7%，超过了人类的能力  
(96%)

**AIoT**

AIoT已经走进企业的核心生产系统，开始创造更大的价值



## 行业案例：新能源车三电分析



### 三电分析应用

反馈真实能耗状态

SOC分析

合理规划

驾驶行为分析

电池、电机、电控技术是新能源汽车的核心技术。使用华为云AIoT方案打造新能源车三电分析方案，挖掘数据价值，为新能源车厂商提供整车工况、电机工况、电池工况、驾驶行为等分析报告，助力新能源车提升整车、电池、电机、电控的设计。

# 赛事案例：基于工业视觉的泡面质检流水线 —— 浙江大学Smartlink俱乐部

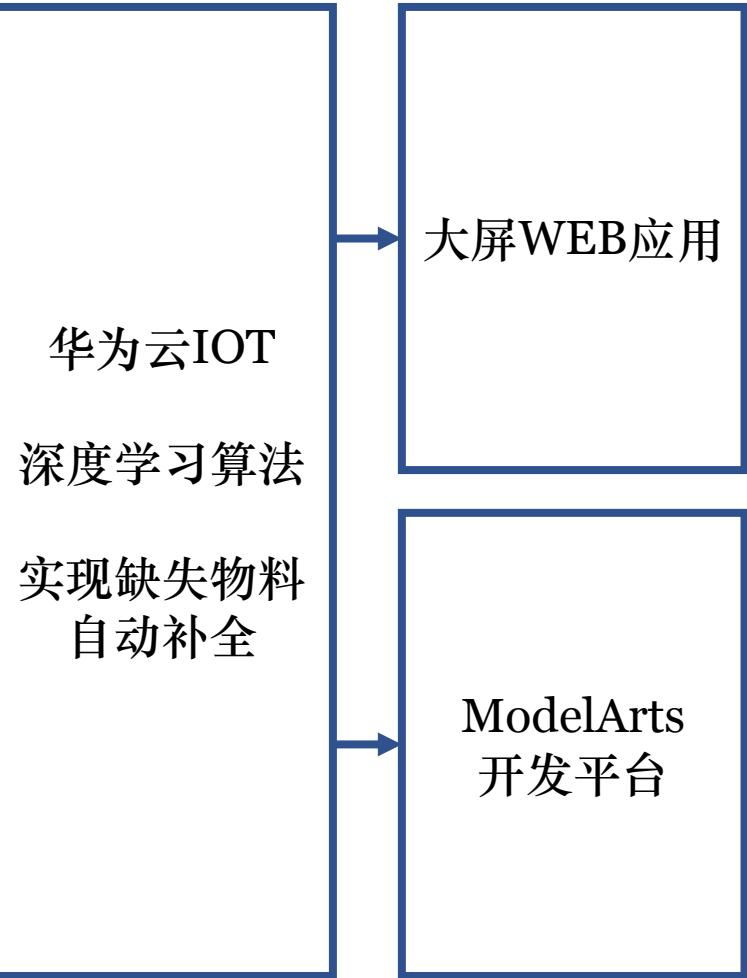


基于工业视觉的自动化泡面质检流水线  
浙江大学SmartLink智联俱乐部钛比小队



## 识别结果

- |   |    |
|---|----|
|  | 缺失 |
|  | 正常 |
|  | 缺失 |



# 从感知、互联到万物智能，AIoT 技术和行业数字化转型与时俱进

## 行业数字化 转型阶段

## AIoT 技术

### 信息化 Industry Informatization

- 通过传感、IT等技术优化行业管理流程，提高信息传递效率。
- 减少人工操作错误，实现内部信息高效整合与利用。

### 数字化 Industry Digitalization

- 对设施设备、业务流程、产品服务等信息全面采集与分析。
- 降本增效优化体验，支撑数据驱动的决策制定。

### 智能化 Industry Intelligence

- 在数字化基础上，加入AI技术，降低对人工的依赖。
- 提高决策的智能化与自动化，加速产品与服务的迭代与创新。



### 万物感知 Sensing

- 机器视觉
- 融合感知
- 空间定位
- SLAM
- 多模态感知
- 语义分割
- 点云配准

### 万物互联 Connecting

- 接入协议
- 网络通信
- 泛协议接入
- 无源物联
- 云边端协同
- 全要素数采
- 确定性时延

### 万物智能 Intelligence

- 数字孪生
- CPS技术
- 离散事件仿真
- AI Agent
- Graph RAG
- 时空模型
- 混合AI

## 让“物”说话



感知物理世界，变成数字信号

- 万物可感知：温度、空间、五觉
- 万物数据化：传感器、物OS

### 关键挑战

- 大量现存设备未被数字化
- 多样化设备、各种行业协议、各种通信网络
- 设备要求低功耗、低成本
- 设备安全问题突出

## 让“物”说一种话



将数据变成Online，提供标准化数据

- 广联接：多协议接入
- 数据标准化：统一物模型

### 关键挑战

- 百万种数据元素，缺乏标准和规范
- 多样化数据格式
- 各厂商烟囱式发展，数据孤岛林立，数据互通困难

## 让“物”说有价值的话



需要解决海量数据处理和分析

- 海量数据分析算力：实时性
- 行业数字化：数字化行业

### 关键挑战

- 海量数据带来算力、存储问题
- 实时流、时序、离线、多维分析复杂
- 行业应用场景的复杂性
- 商业闭环、价值变现和分配问题

## 行业观察：众多企业发力 AIoT 业务

Haier



阿里云



联通数码  
Unicom Digital Tech



HIKVISION



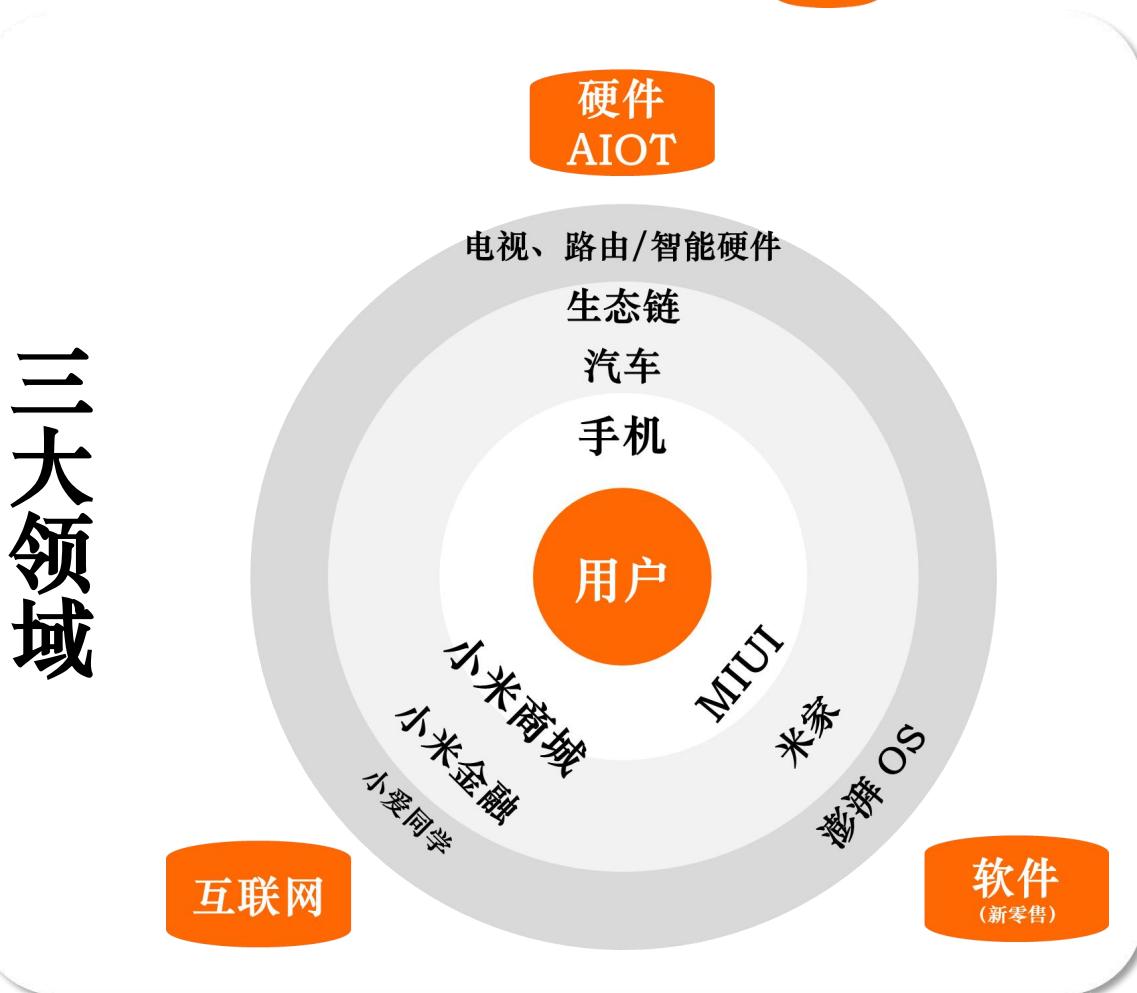
Baidu 百度



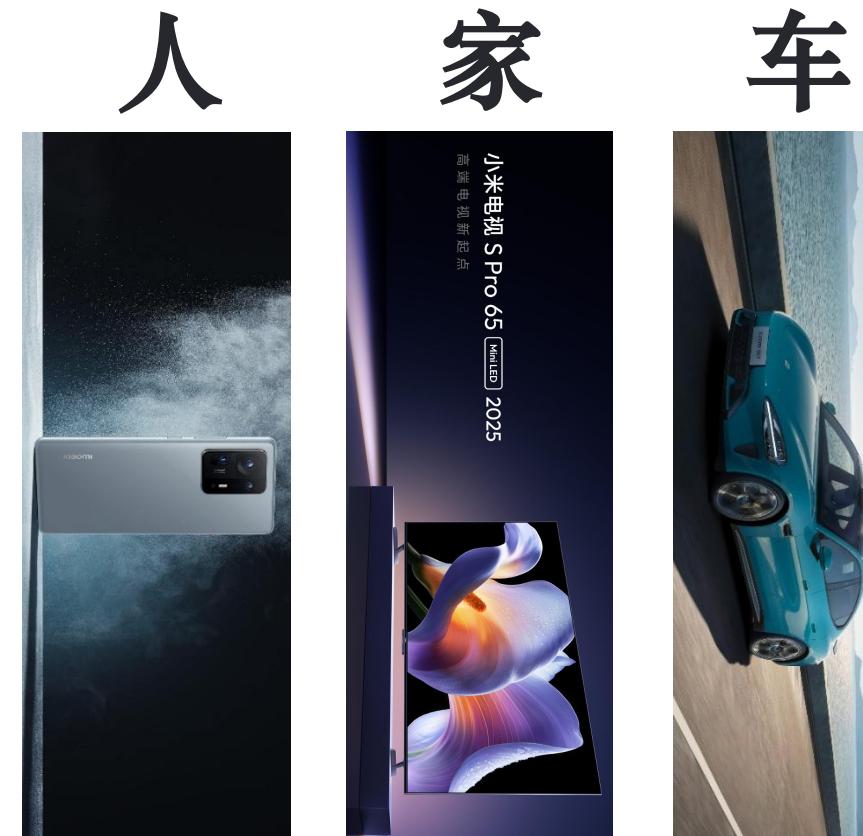
Rockchip  
瑞芯微电子

TCL

# 大厂动向：Xiaomi



## 三大场景



# 大厂动向：HUAWEI



基于华为云，打造云边端协同



华为云+鸿蒙：  
快速构建安全系统



Huawei Proprietary - Restricted Distribution

资料来源：华为云采访资料

## 大厂动向：**HIKVISION**

### 三大业务群

公共服务事业群（PBG）  
企事业事业群（EBG）  
中小企业事业群（SMBG）

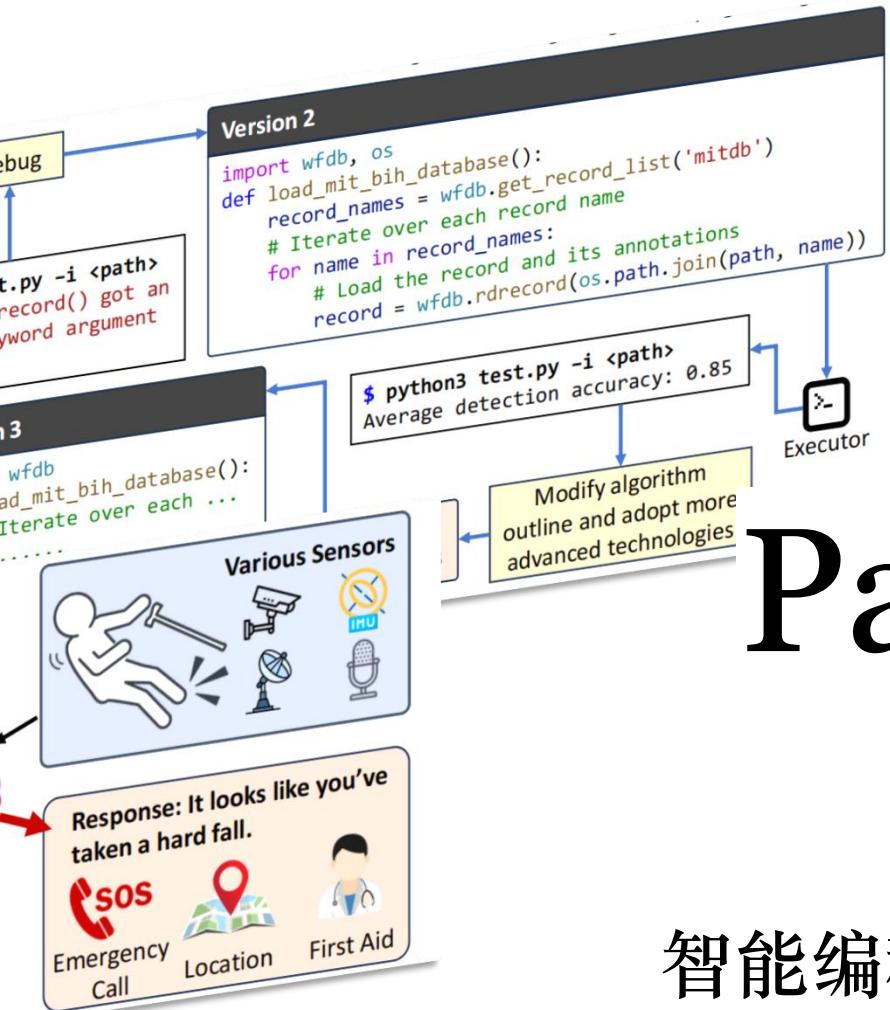
### 三大产品家族

边缘节点产品：节点全面感知  
边缘域产品：域端场景智能  
云中心产品：中心智能存算



小组成员前往杭州海康参观图片

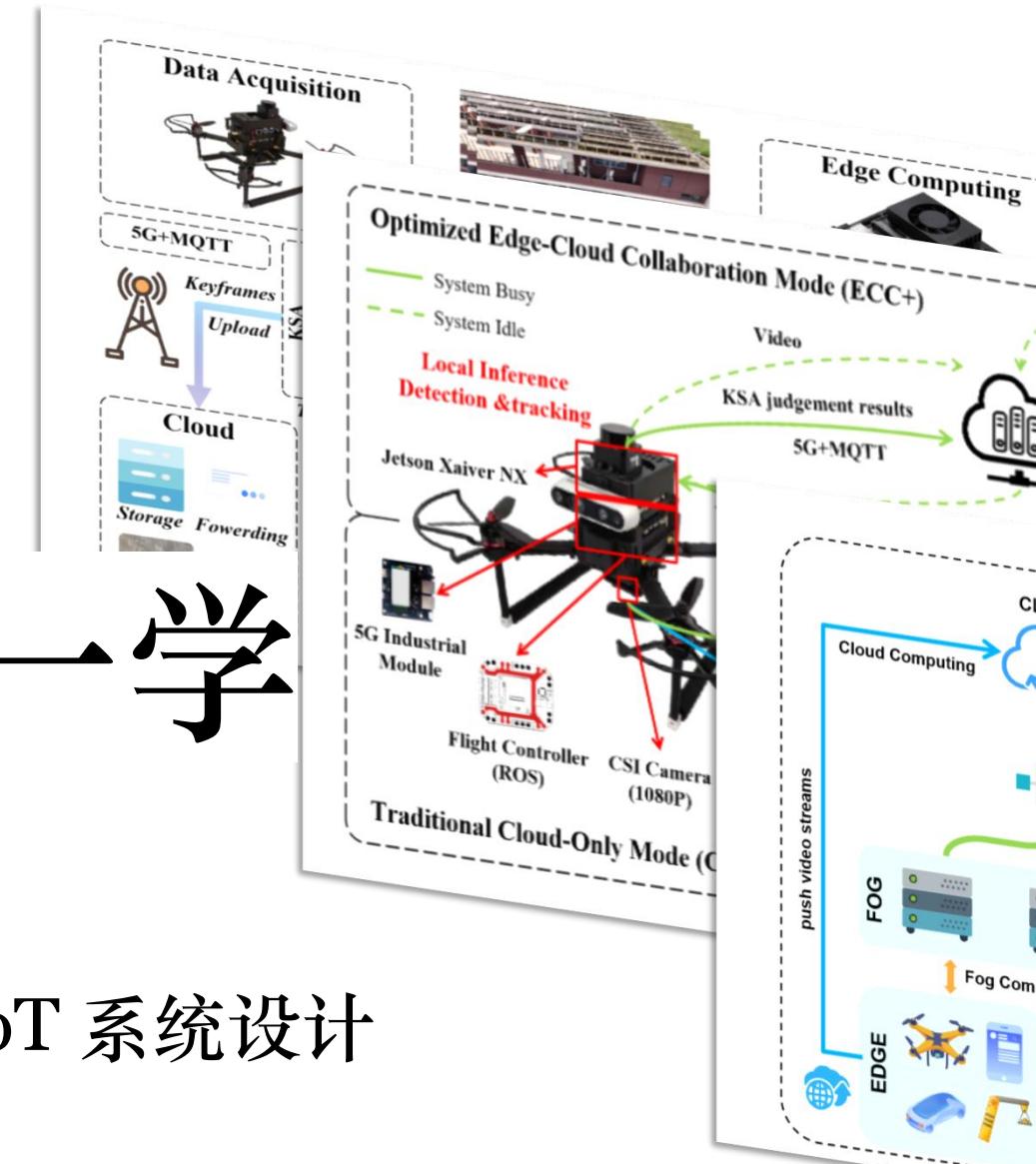
作为智能物联领域的“视觉中枢”，通过其独特的“感知+AI+数据”三位一体架构，为物理世界打造了数字化的映射。公司自研的物联网平台“海康云眸”已成功连接全球超过1.5亿的智能设备，构建了一个全面的感知网络



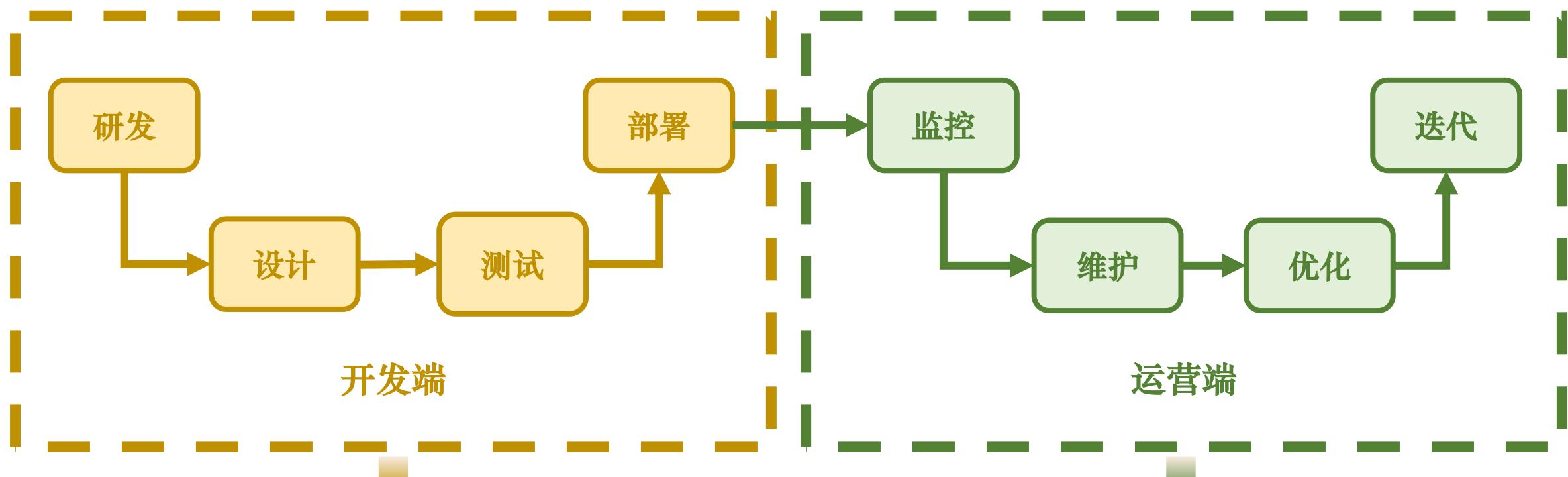
## Part 2

学 — 学

# 智能编程与高效部署的 AIoT 系统设计

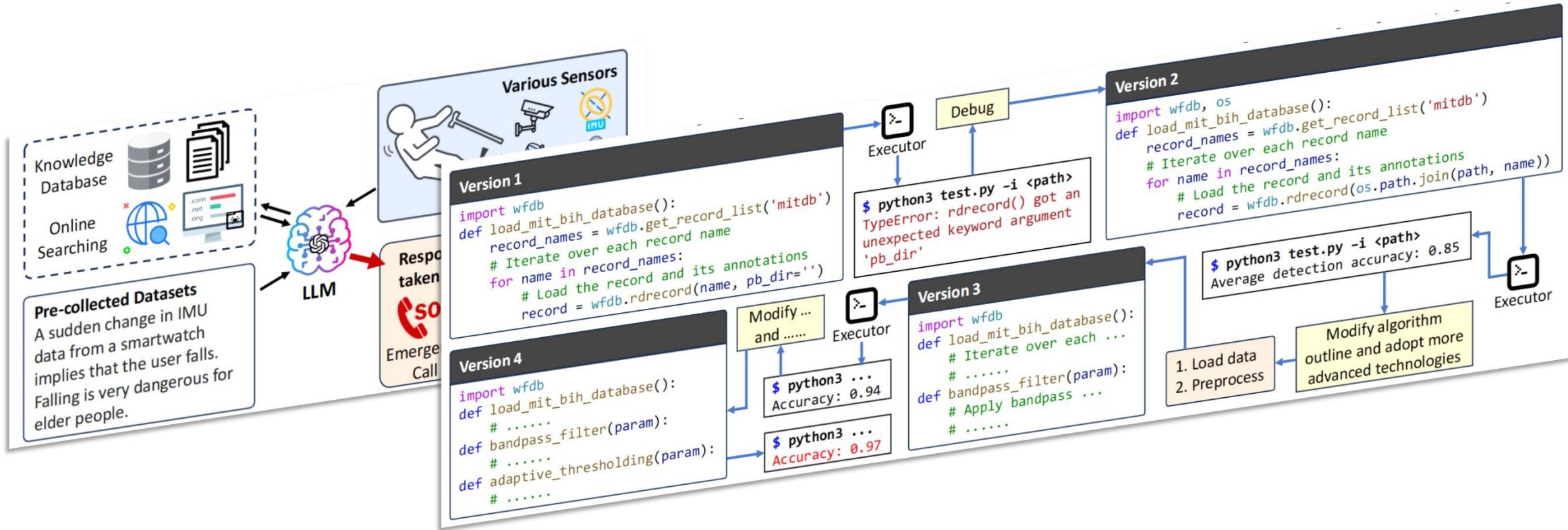


# 产品的全生命周期



AIoT 开发范式的智能化演进

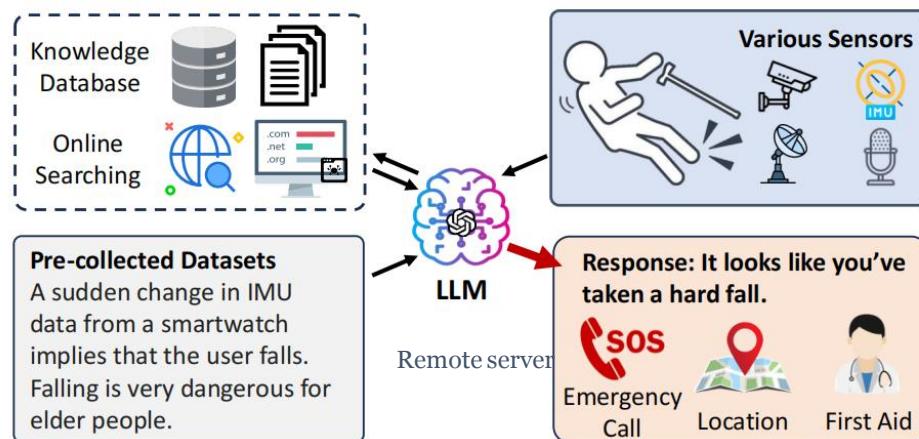
系统部署与推理的实时性与协同性增强



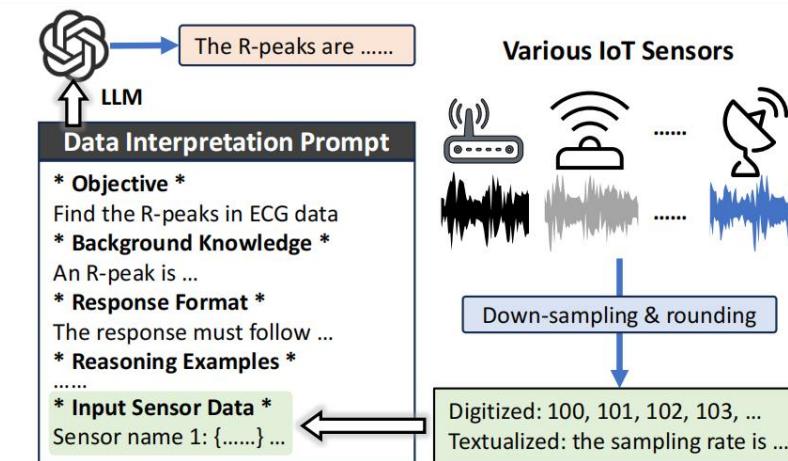
# AutoIOT: 基于大语言模型的 AIoT 应用自动化自然语言编程

# Background

现代 LLM (如 GPT-4) 不仅能处理自然语言, 还具备处理结构化数据的能力, 这就为直接处理如IoT中传感器等的数据提供了可能(Penetrative AI)



Application of Penetrative AI

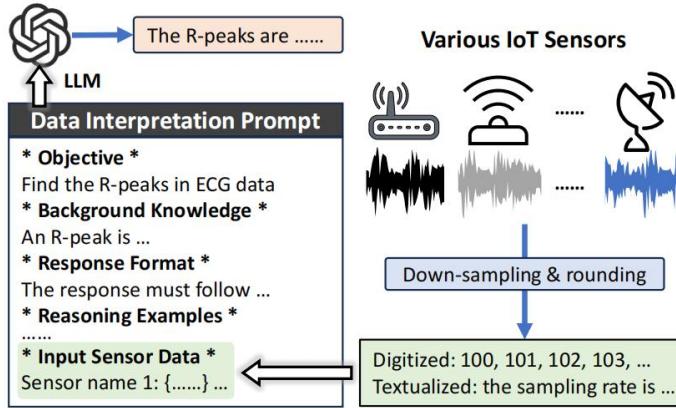


You can see from here, traditional methods always pack up all the data to LLMs and conduct programs remotely which users can't detect

- **黑箱推理:** LLM 内部推理过程对用户不透明, 难以验证结果的正确性。
- **隐私与通信:** 将原始或中间传感器数据传输到远端服务器存在隐私泄露、费用高和延时问题。
- **数据长度与 token 限制:** 传感器数据通常量大且高维, 直接传输受限于 token 数量, 下采样导致性能下降。

## Weakness

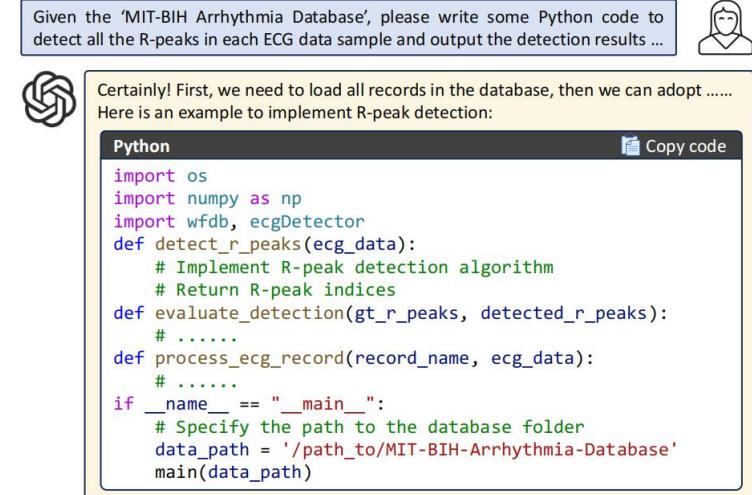
# Key motivations



通过本地LLMs生成代码或程序

直接了解数据处理过程  
无需担心token与数据传输丢失问题

Given the 'MIT-BIH Arrhythmia Database', please write some Python code to detect all the R-peaks in each ECG data sample and output the detection results ...



Certainly! First, we need to load all records in the database, then we can adopt ..... Here is an example to implement R-peak detection:

**Python**

```

import os
import numpy as np
import wfdb, ecgDetector
def detect_r_peaks(ecg_data):
    # Implement R-peak detection algorithm
    # Return R-peak indices
def evaluate_detection(gt_r_peaks, detected_r_peaks):
    # .....
def process_ecg_record(record_name, ecg_data):
    # .....
if __name__ == "__main__":
    # Specify the path to the database folder
    data_path = '/path_to/MIT-BIH-Arrhythmia-Database'
    main(data_path)

```

**Copy code**

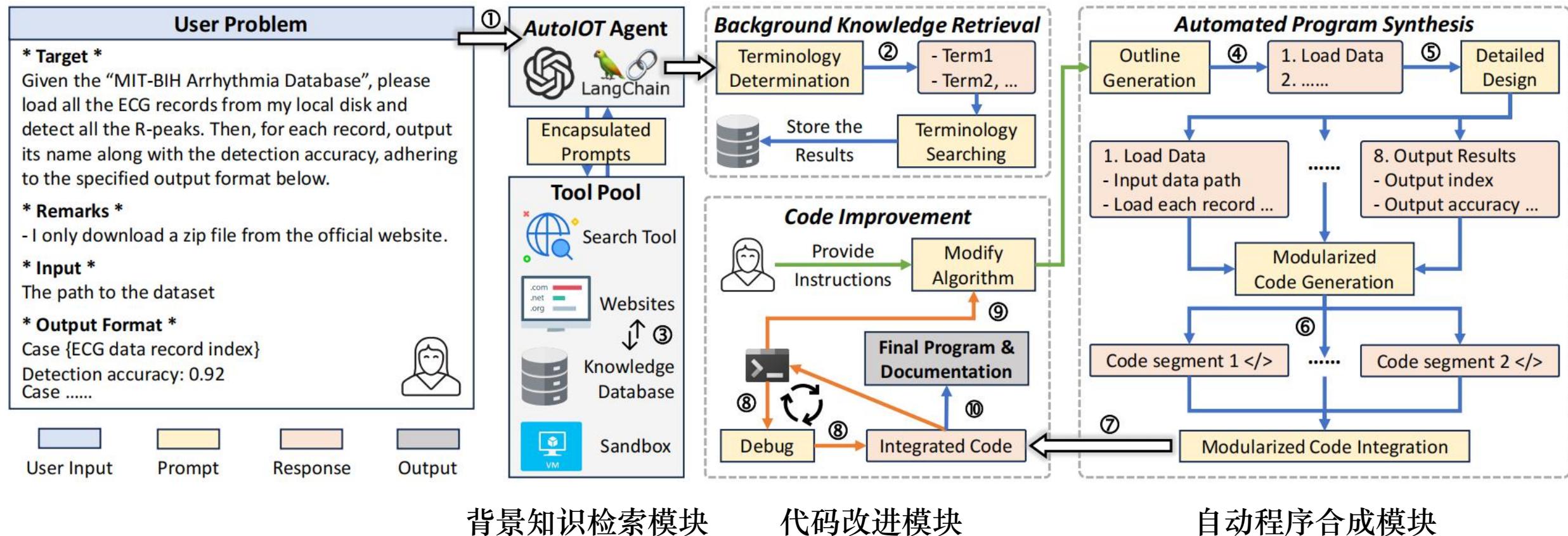
## Unlucky, there are still some challenges

- 代码生成不充分：往往仅能生成空函数或高代码框架，无法给出完整的实现。
- 领域知识缺失：对最新、专业的 AIoT 算法和数据处理流程可能不熟悉。
- 需要大量用户干预：用户往往需要反复提供反馈和手动调整。



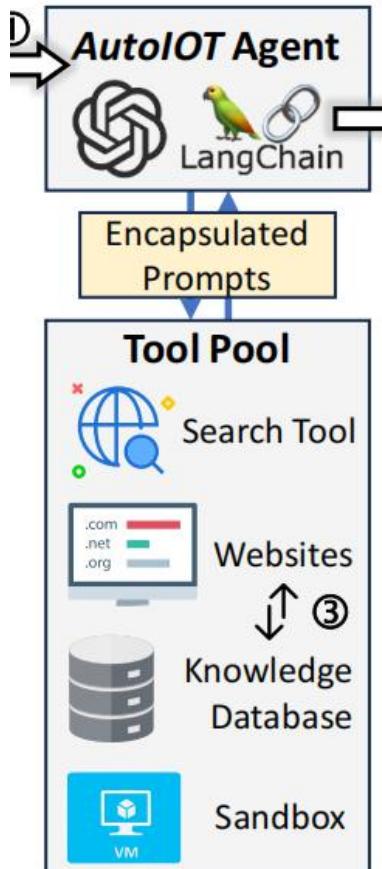
## The proposal of AutoIOT

# System Design



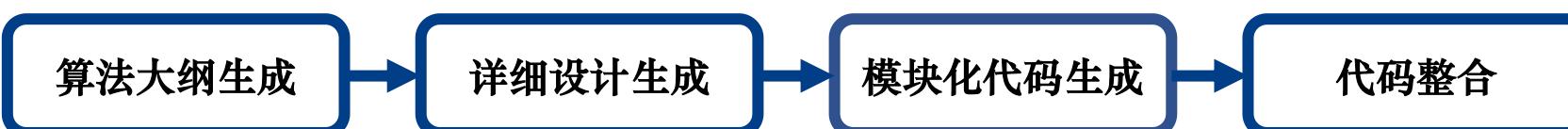
利用 **LangChain** 框架将网络搜索、知识库检索和代码执行等工具集成到 LLM 的对话过程中  
使得整个流程能够自动调度、动态交互

# System Design ——背景知识检索模块



- **功能：**自动识别用户需求中的关键术语，并通过网络搜索（例如从 Wikipedia、GitHub 等）获取最新的领域知识和相关定义。
- **处理方式：**将检索到的网页内容转换为向量表示（利用文本嵌入技术），建立一个本地的知识库，以便在后续的上下文中提供相关信息支持。

# System Design —— 自动程序合成模块

- 核心思想：利用链式思维（Chain-of-Thought, CoT）提示，分步骤引导 LLM 生成算法大纲、详细设计以及各个模块的代码。
- 工作流程
 

```

graph LR
    A[Algorithm Outline Generation] --> B[Detailed Design Generation]
    B --> C[Modular Code Generation]
    C --> D[Code Integration]
  
```

Terminology Determination
<b>* User Problem *</b> <user_input>...</user_input>
<b>* Motivation *</b> I need to understand the meaning of some concepts in the user's problem to gain some background knowledge.
<b>* Target *</b> Based on the user's problem, please determine a list of terminologies that I need to search online. Please only output a list of terminologies.
<b>* Response Format *</b> term1, term2, ...

(a) Terminology determination

Terminology Searching
<b>* User Problem *</b> <user_input>...</user_input>
<b>* Target *</b> Use the web search tool to search for precise definition or description of <b>{terminology}</b> .
<b>* Rules *</b> - Wikipedia is preferred. - Filter out the contents that are irrelevant to the problem. - Do not provide algorithms or implementation details.
<b>* Response Format *</b> URL1, URL2, .....

(b) Terminology searching

Algorithm Outline Generation
<b>* User Problem *</b> <user_input>...</user_input>
<b>* Target *</b> Based on the background information in the context documents, please provide an algorithm outline step by step to solve the user's problem.
<b>* Rules *</b> - Use the web search tool multiple times. - Analyze the retrieved information and filter out irrelevant contents. - Refer to the context.
<b>* Response Format *</b> Step 1: [title] - xxx Step 2: xxx

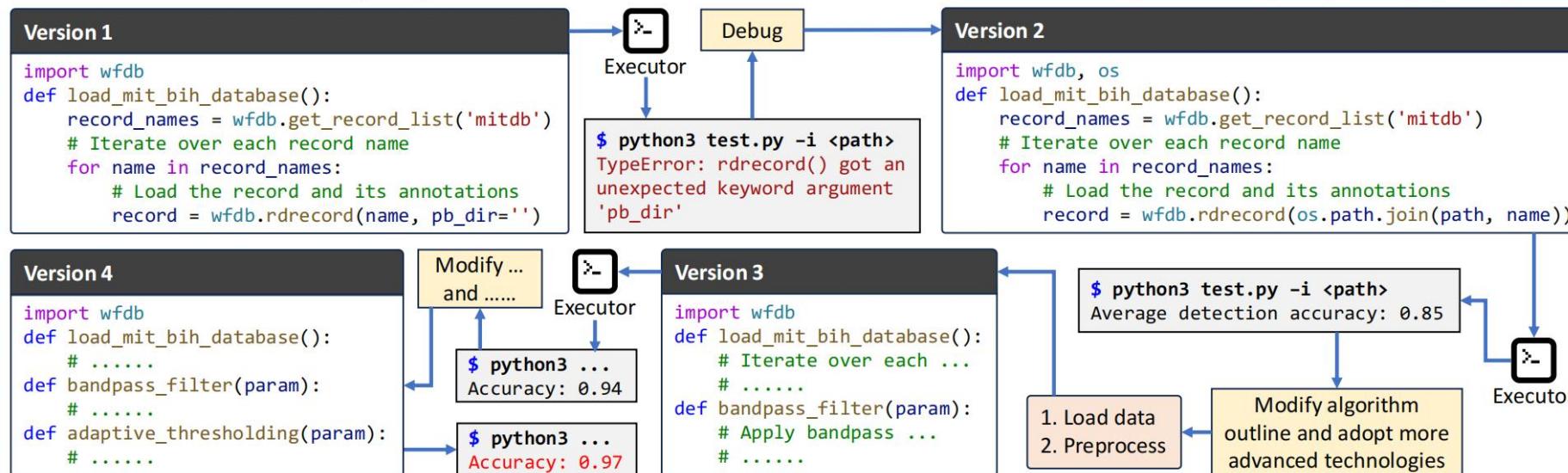
(c) Algorithm outline generation

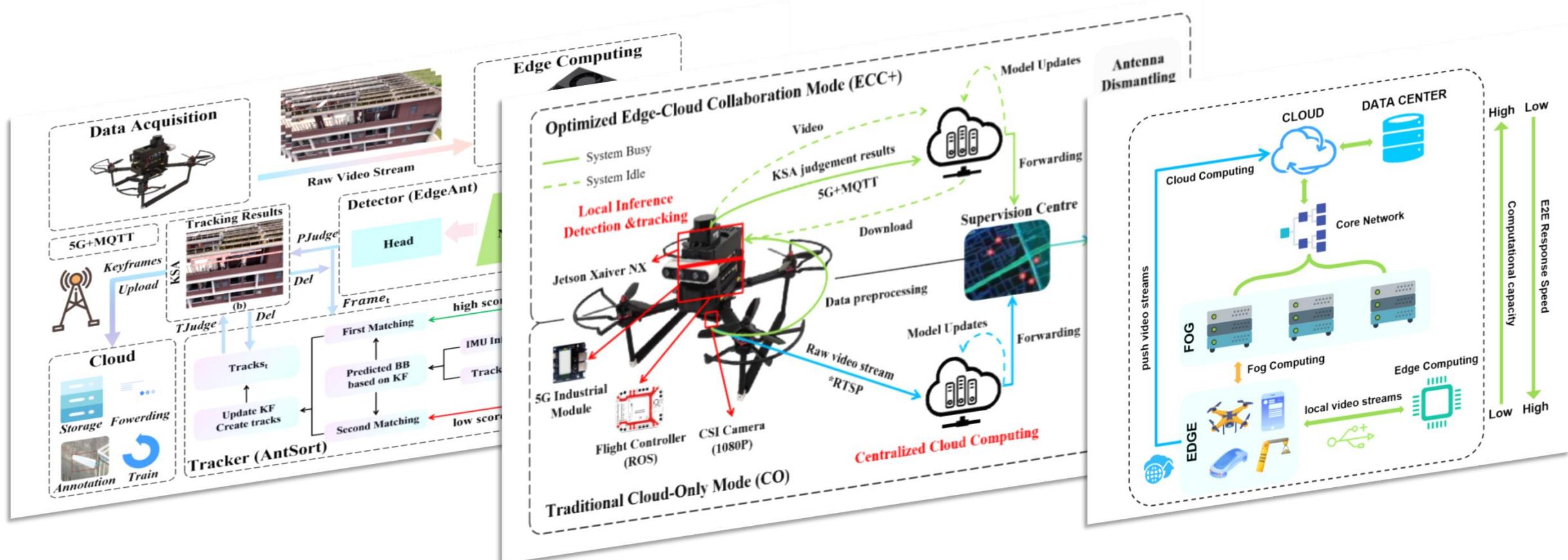
Detailed Design Generation
<b>* User Problem *</b> <user_input>...</user_input>
<b>* Target *</b> Please elaborate on each step with detailed technologies or algorithms.
<b>* Rules *</b> - Do not modify the outline. - Use the web search tool. - Filter out irrelevant content.
<b>* Response Format *</b> Step 1: [title] - xxx Step 2: xxx

(d) Detailed design generation

# System Design —— 代码改进模块

- **自动调试:** 将生成的代码在沙箱环境中执行，捕获编译器或解释器返回的错误信息，并将这些反馈嵌入提示中，指导 LLM 进行迭代修正。
- **性能优化:** 除了基本的错误修复，还可以利用网络搜索等工具，引入更高级的算法来改进程序性能（例如在心跳检测中采用更精细的信号预处理方法）。
- **迭代更新:** 通过多轮交互和反馈循环，使得生成的程序不断完善，达到用户需求规定的精度和效率要求。

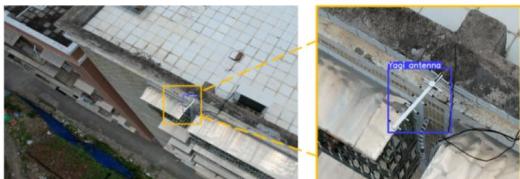




# Real-Time AIoT for UAV Antenna Interference Detection via Edge-Cloud Collaboration

# Background

居民未经授权安装天线 → 同一区域频谱拥堵，影响基站稳定



几种常见的未经授权的居民天线

无人机进行天线巡逻

5G: 超高带宽 + 可控延迟

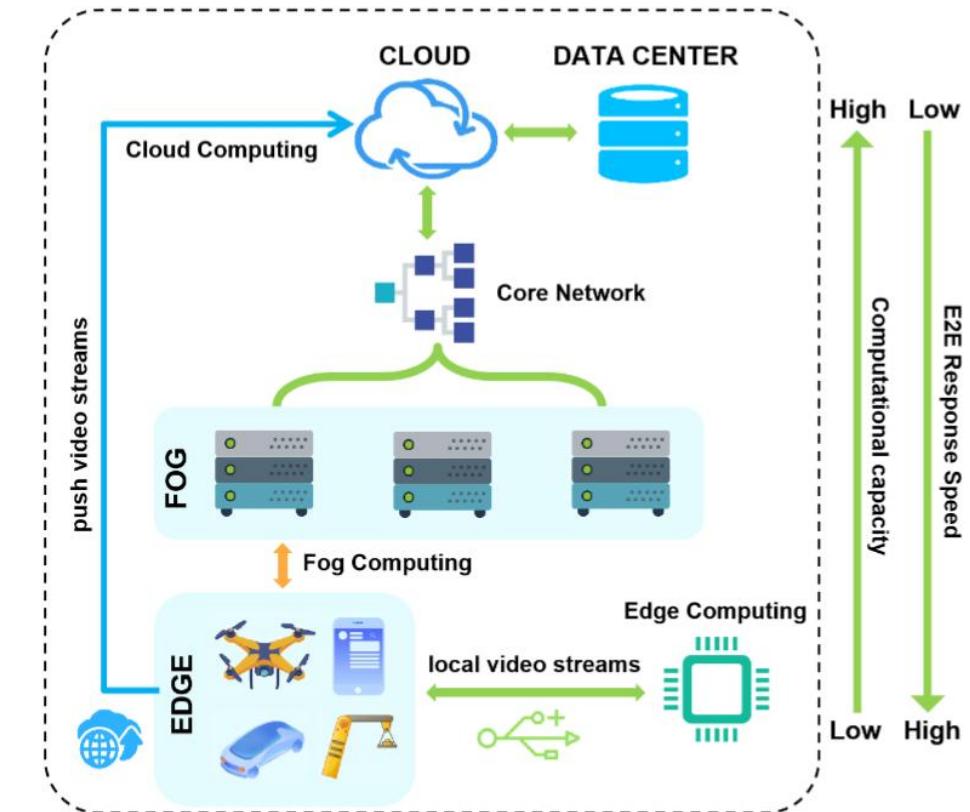
大幅提高工作效率，实现大规模干扰源检测

边缘计算和物联网设备的快速发展

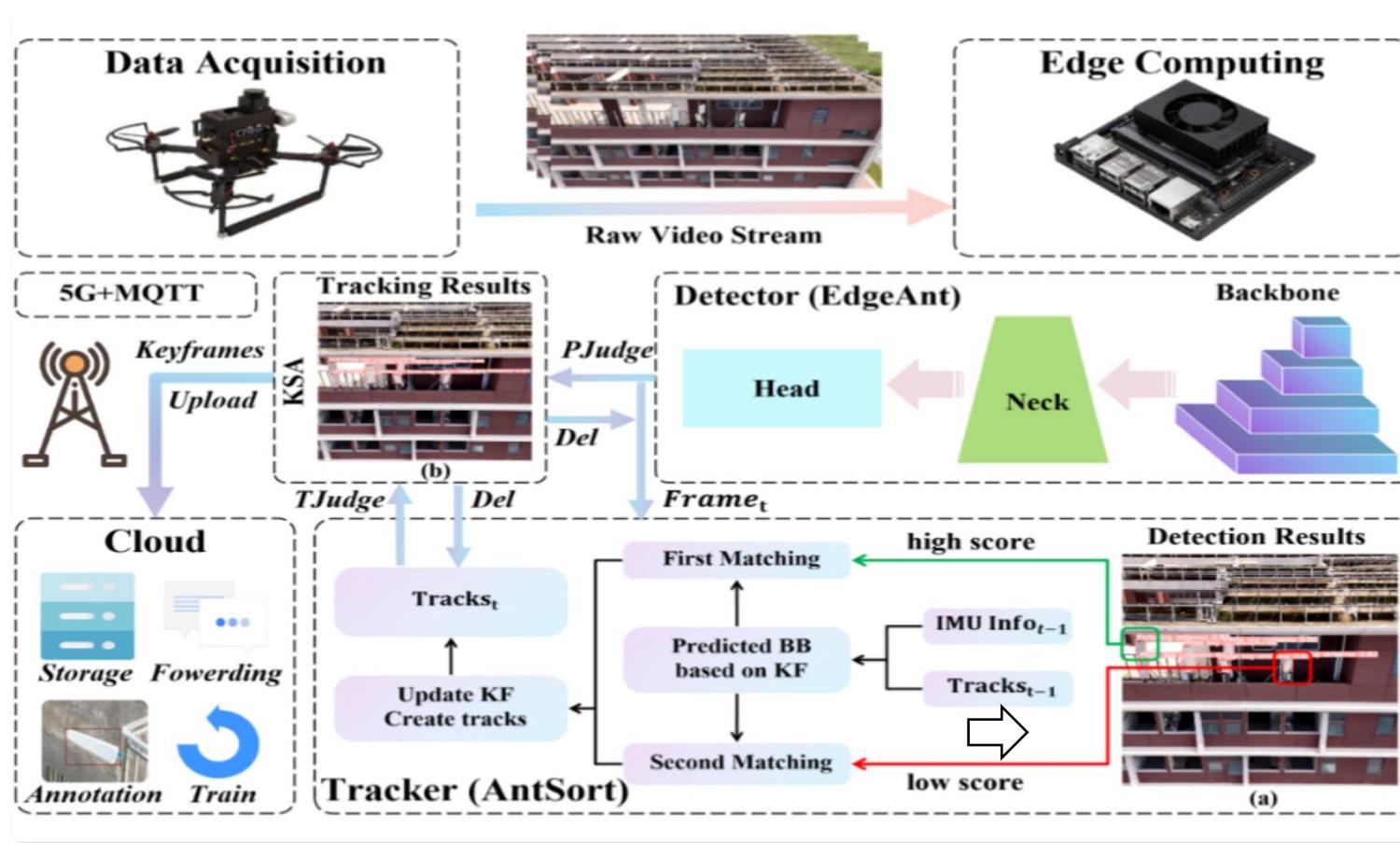
利用边-云协同(ECC+)的模式

通过无人机(UAV)进行天线干扰源检测

解决5G通信当中出现的未经授权或故障天线引起的通信干扰问题



# Methods



系统工作流水线

基于计算机视觉算法、边缘-云协作模式  
构建了一套先进的AIoT识别检测系统

计算机  
视觉模型

优化  
边-云协作模式

关键帧  
选择算法

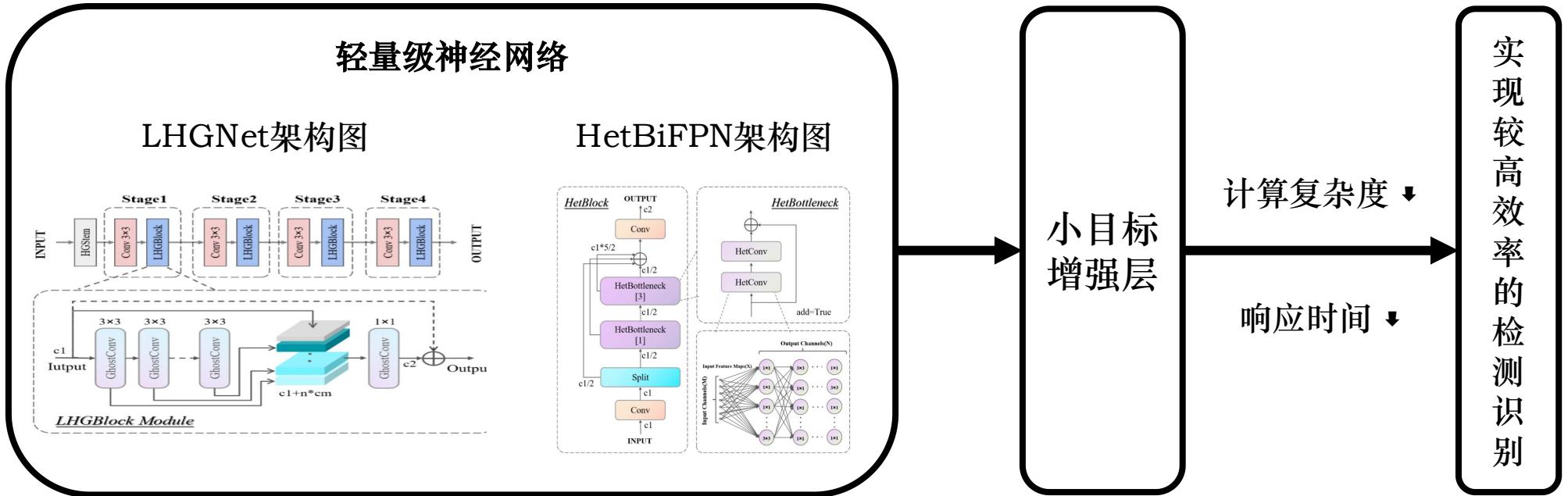
端到端延迟下降  
可靠性提升

实现更好地对违规天线的识别和判断



# 核心天线定位检测方案

EdgeAnt  
轻量级检测器



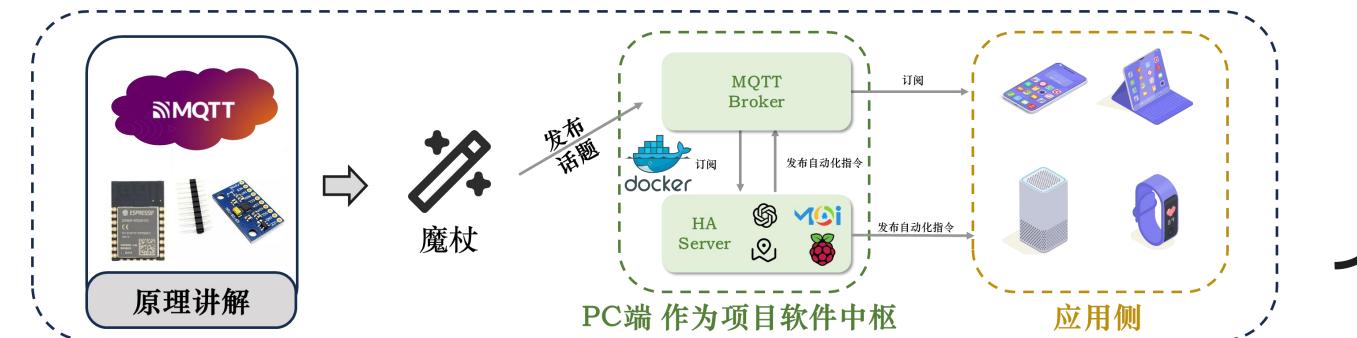
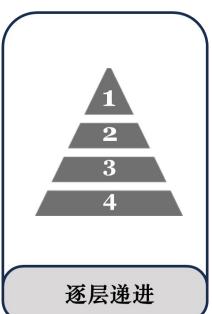
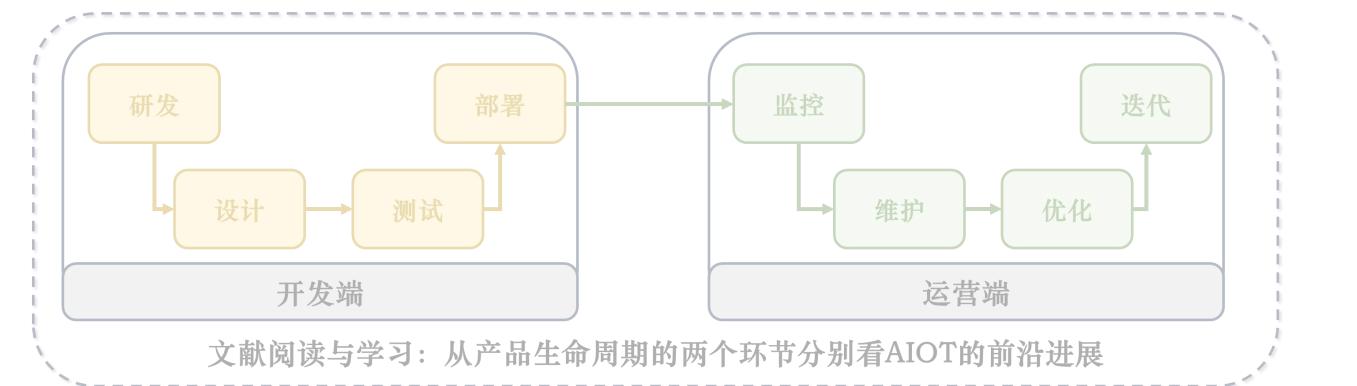
AntSort  
检测器

无人机惯性测量单元

引入运动误差补偿

提高跟踪的准确性和鲁棒性

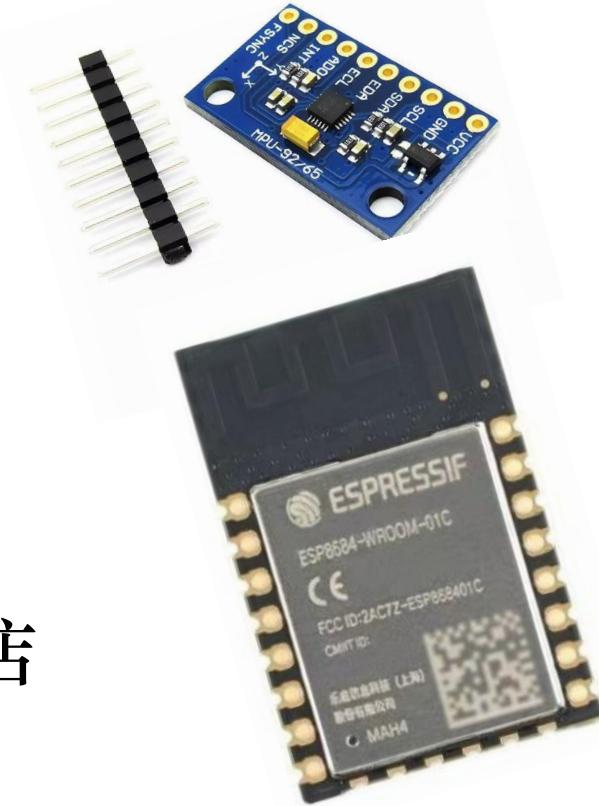
针对干扰源检测任务优化的跟踪器



# Part 3 做一做

Home Assistant + MQTT + HACS插件商店

魔杖化身家庭物联网中枢

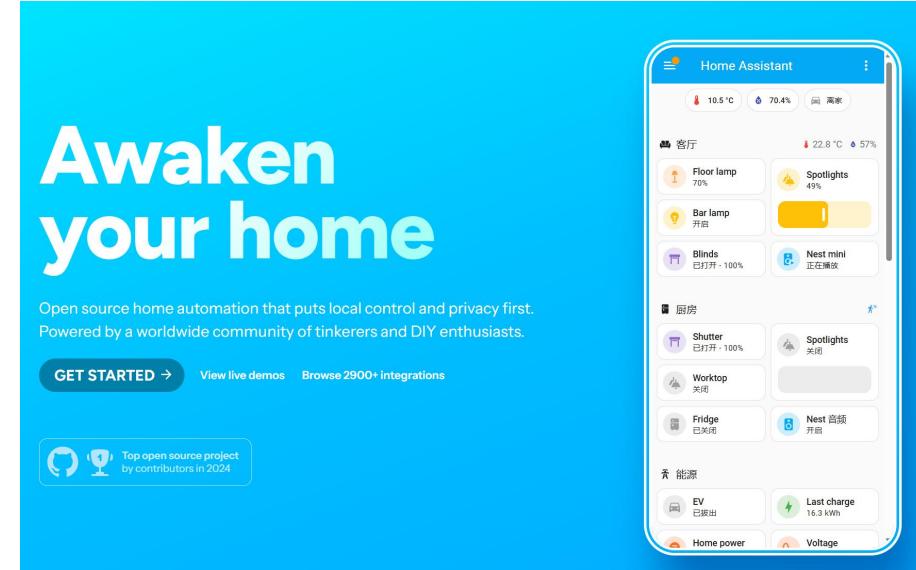




# MQTT

MQTT (Message Queuing Telemetry Transport) 是一个轻量级的消息发布/订阅协议，特别适合资源受限设备（如传感器）和低带宽、不可靠的网络环境。

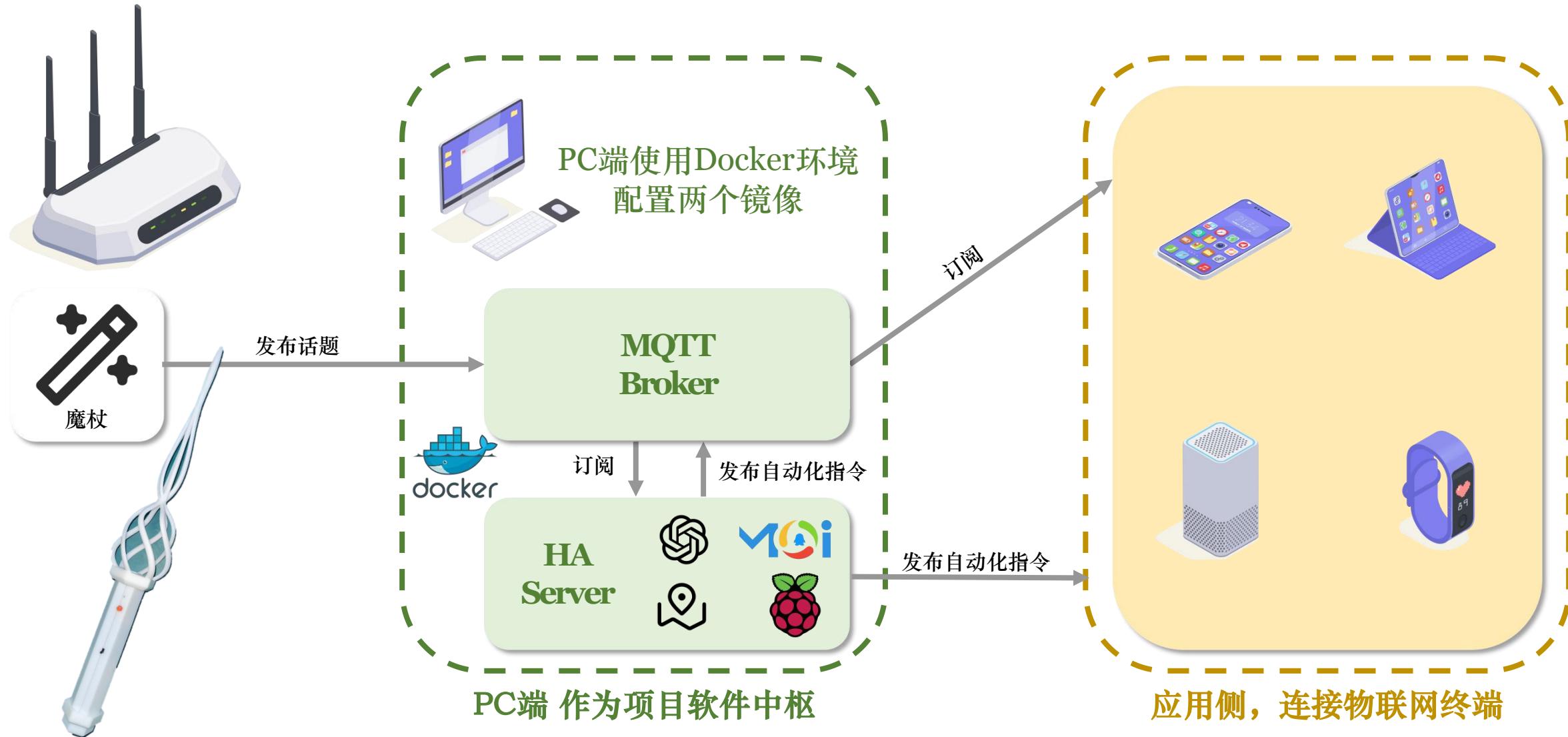
- **轻量级**
- **发布/订阅模式**
- **实时性强**: 支持实时消息传输。
- **广泛支持**



# Home Assistant

Home Assistant 是一款开源的家庭自动化平台，旨在帮助用户将各种智能设备集成到一个统一的界面中进行控制和管理。它具有以下特点：

- **广泛的设备支持**: 支持数千种智能设备和服务。
- **自动化**
- **本地化和隐私**
- **扩展性**: 通过社区开发的插件和集成



# 核心部件



ESP32



MPU6050

带Wi-Fi 和 蓝牙模块

三轴加速度计和三轴陀螺仪的传感器

# 数据采集

MPU6050每次采样会产生6个数值(三轴加速度和三轴角速度)

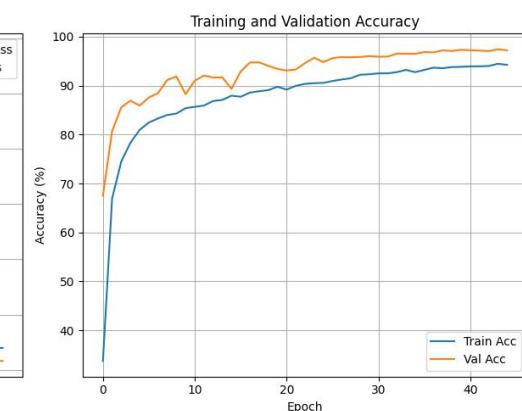
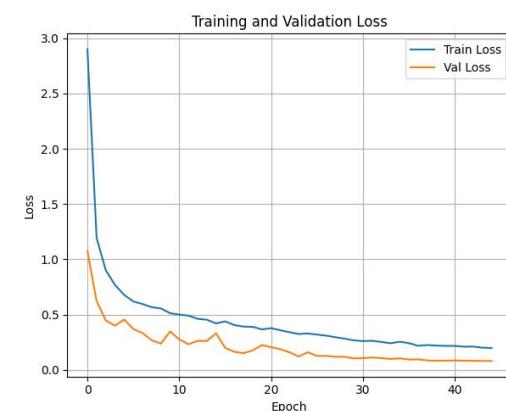
采用线性插值的方法，将每个动作序列重采样为固定长度(默认100帧)

将所有数据缩放到同一区间，使得各个特征在训练中具有相同的重要性

# 模型训练

使用有两个卷积层、一个全连接层带有dropout的CNN  
使用 lr=0.001，交叉熵损失函数，训练了50个epoch

```
def __init__(self, input_shape, output_classes):
    super(ActionClassifierCNN, self).__init__()
    # 卷积层
    self.conv1 = nn.Conv2d(in_channels=input_shape[2], out_channels=30, kernel_size=3,
                         stride=3, padding=1)
    self.conv2 = nn.Conv2d(in_channels=30, out_channels=15, kernel_size=3, stride=3,
                         padding=1)
    # 全连接层
    self.fc1 = nn.Linear(15 * ((input_shape[1]-1)//9+1), output_classes)
    self.dropout = nn.Dropout(0.3)
```





# 动作识别测试

EMQX Dashboard

localhost:18083/#/websocket

## WebSocket 客户端

客户端 ID: emqx\_ODMOND | 用户名: wand | 密码: \*\*\*\*

Keepalive: 60 | Clean Start: true | 会话过期时间: 0

TLS: false | 协议版本: MQTT 5

### 订阅

主题: magicWind/control | QoS: 0 | 订阅

主题	QoS	时间	操作
magicWind/control	0	2024-12-20 17:05:57	取消

### 发布

主题: magicWind/control | Payload: {"msg": "hello"} | QoS: 0 | Retain:

### 已接收

主题	QoS	时间
暂无数据	暂无数据	暂无数据

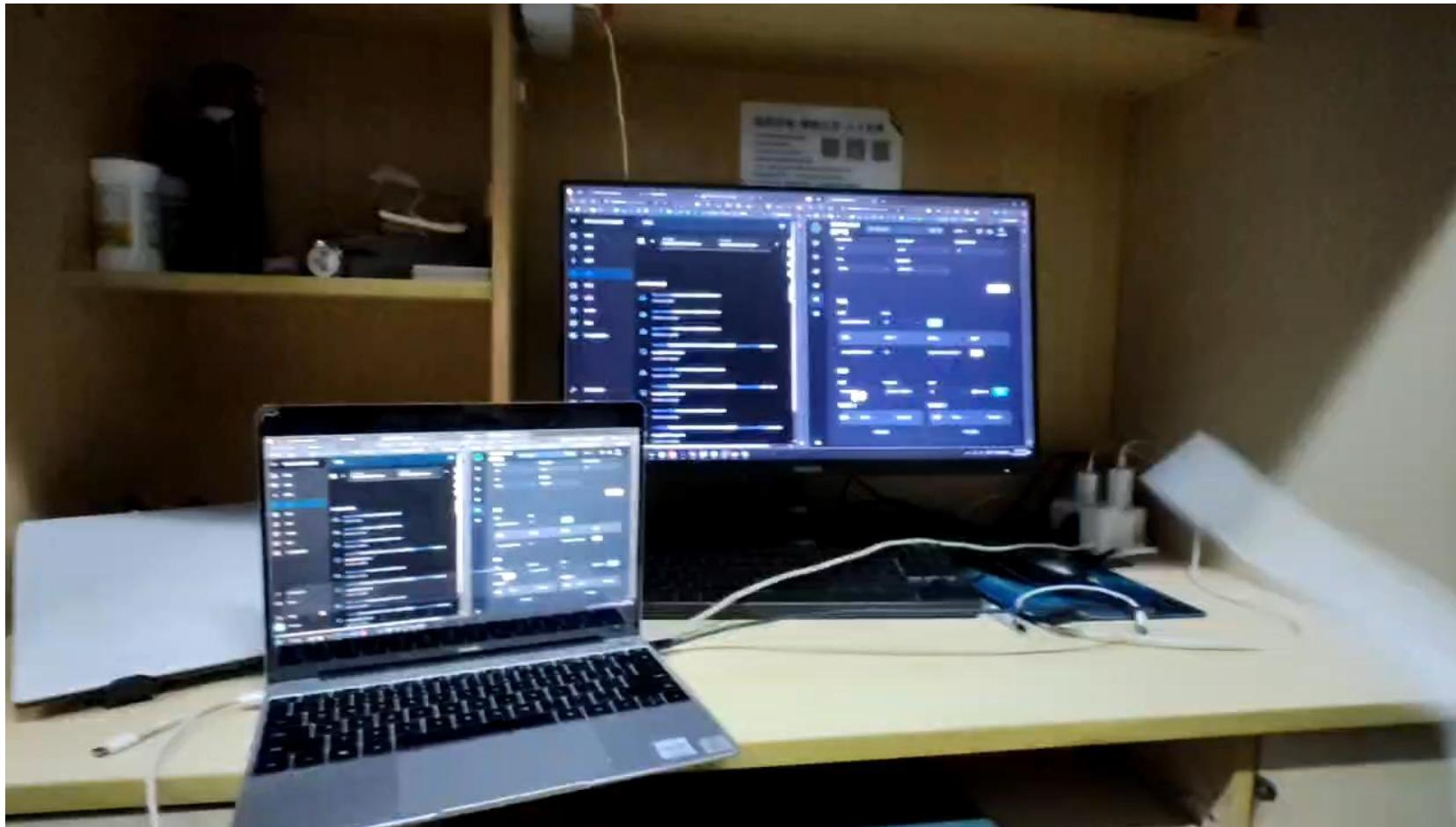
### 已发送

主题	QoS	时间
暂无数据	暂无数据	暂无数据

# 手机实拍



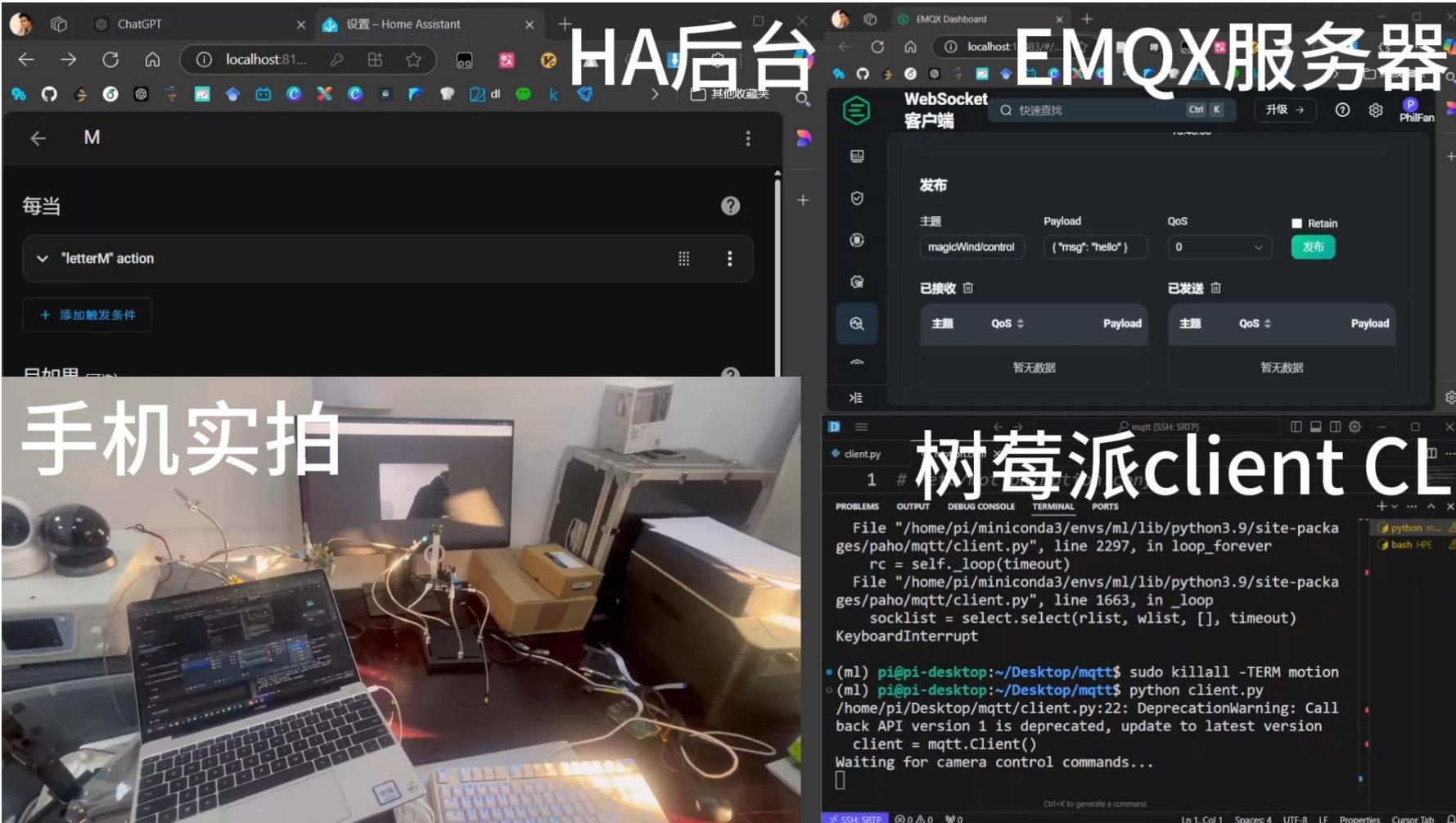
## 使用HA系统+魔杖控制小灯



这里我们使用“上  
下”这个动作

作为触发开关灯的  
关键词

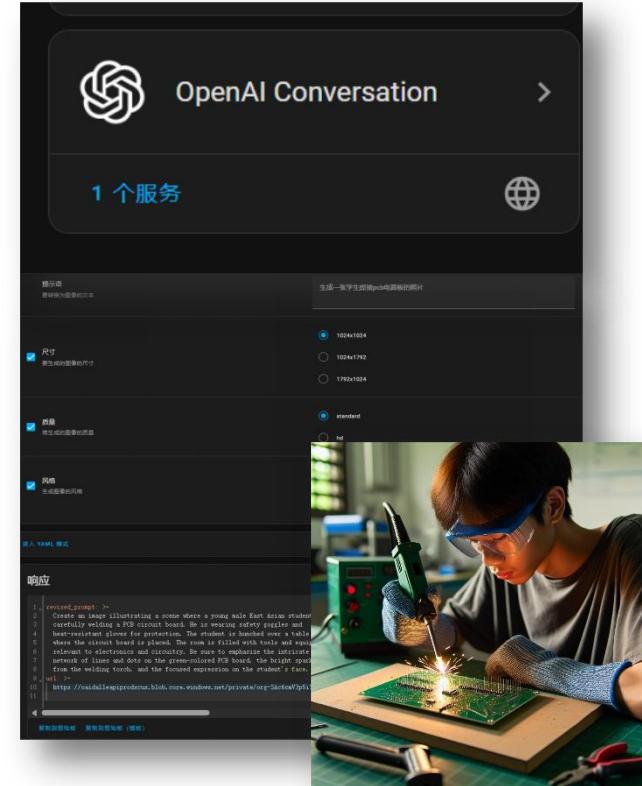
# 使用HA系统解决SRTP项目中的一个实际问题



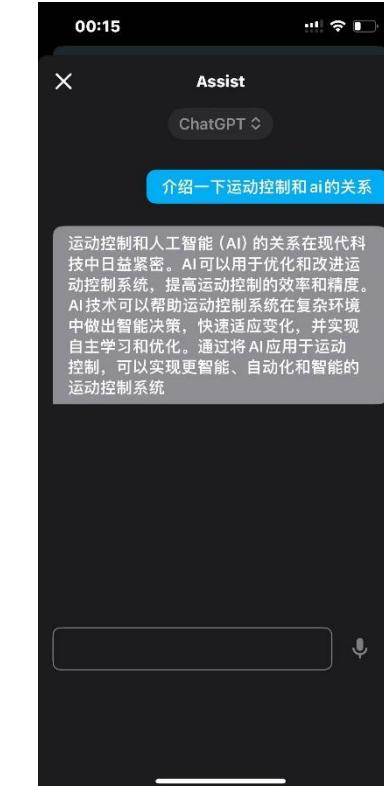
# 接入大模型，为家庭智能开拓一个DIY方向



支持接入文生图等多种api



多端同步，语音助手

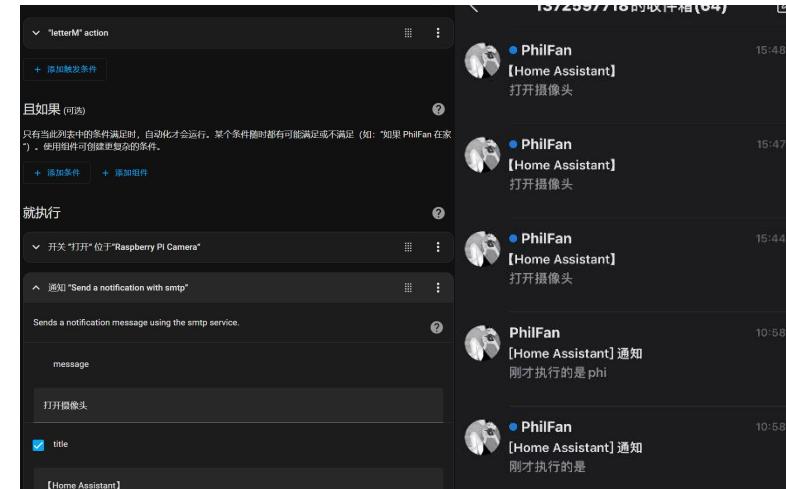


# What's More

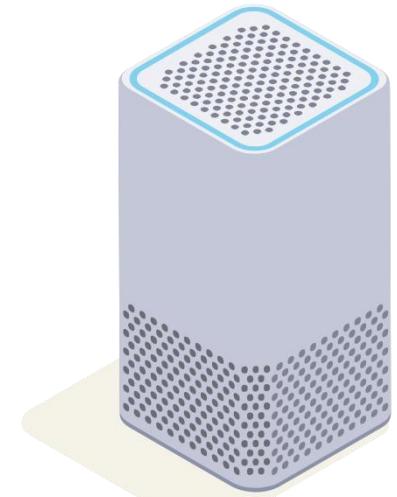
利用Home Assistant 还可以完成更多



集成连接互联网摄像头  
(如上面的萤石和乐橙)



使用SMTP服务，在打开摄像头的同时，发送邮件



集成HomePod，小爱同学  
等智能音箱当tts  
(写了相关程序，但是没有对应硬件)