

# A14 基于 Agent 的流程工业建模数据

## 智能优选与闭环寻优系统

### 项目详细方案

2026 浙江省服务外包创新创业大赛 · 企业命题【和利时】

命题方向：智能计算 | 题目类别：应用类 | 产品名：ProcessPilot 工业辨识数据智能优选工作台

开发栈：Python + LangGraph + FastAPI + Vue 3 | 源码版本：HEAD=v0.1.43

本方案严格对照赛题《任务要求》的 7 项任务与《技术要求与指标》的 5 大模块逐条展开；所有实现细节锚定源码（v0.1.43，133 项判定其中 86 项带代码锚点可追溯），全部数字来自探针战役 / 压力测试 / 安装态实测。论证采用答辩思维导图口径，诚实边界主动披露。

## 第 1 章 项目概述

### 1.1 命题背景

在流程工业（化工、炼化、电力、冶金、建材、制药、食品、造纸等）的数字化转型中，先进控制（APC）已成为企业实现节能降耗、提质增效的核心手段。然而 APC 策略的成功部署，高度依赖底层被控对象数学模型的辨识精度；而建模的成败，从根本上取决于输入数据的质量。面对真实工业场景中伴随强噪声、多变量强耦合与复杂时滞特性的时序数据，传统的依赖人工调参、缺乏即时反馈的预处理方式，已成为制约建模效率与精度的技术瓶颈。

### 1.2 公司背景（命题企业）

杭州和利时自动化有限公司成立于 2003 年 9 月，从事智能制造控制系统、工业软件、自动化仪表及工业信息安全相关产品的研发、制造、工程实施与系统服务，面向核电、火电、热电、新能源、炼化、化工、油气管道、冶金、建材、制药、食品、造纸等众多流程工业企业提供智能工厂整体解决方案。公司每年承担各类智能化改造和数字化转型工程项目 8000 多项，是国内领先的智能化系统解决方案供应商，也是国内唯一被国际权威调查咨询机构 ARC 评定为“全球自动化 50 强”的中国企业。

### 1.3 业务背景与痛点

- **有效样本被淹没：**企业采集的海量时序数据中，真正富含动态价值的有效辨识样本，往往被大量无效的稳态数据淹没——工厂 90% 以上时间处于稳态运行。
- **预处理优劣滞后验证：**手动调整预处理参数不仅试错周期长，且预处理的好坏只能在最终建模后才能验证，单次迭代常以天计。
- **难寻全局最优：**割裂的数据筛选与模型验证，使工程师极难寻得全局最优的数据集组合。
- **门槛高：**工业数据分析依赖资深工程师经验，知识难以沉淀与复用，新人上手周期长。

### 1.4 设计目标与原则

- **目标：**把“数据预处理 + 系统辨识 + 策略寻优”这一靠人反复试错的环节，变成一键式自动化流水线，建模周期从“天级”压缩到“分钟级”。
- **原则一 · 定位清晰：**做 APC 前端加速器，不替代 APC，补前置缺口。
- **原则二 · 模块化：**底层算法库高可靠、标准化参数接口，便于 Agent 动态调度。
- **原则三 · 可落地：**直连现场 DCS（只读不写），成果导出为和利时 MPC 组态格式。
- **原则四 · 诚实：**数字如实报、边界主动披露，以可信度取胜。

### 1.5 国内外相关工作

传统的工业数据预处理依赖 Aspen Tech、MATLAB System Identification Toolbox 等商业工具，需工程师逐项手工配置，缺乏反馈闭环；近年出现 talktodata、DeepBI 等对话式数据分析工具，但多面向通用表数据，未针对“系统辨识前数据优选 + 闭环寻优”这一垂直场景深度优化。浙大中控等厂商的 APC 解决方案虽有 LLM 编排与多智能体架构，但缺乏闭环寻优能力。本项目的差异化在于：将自然语言驱动的 Agent 编排与工业辨识算法库深度耦合，并以辨识效果为目标函数实现闭环自寻优。

1.6 赛题要求对照表

| 赛题模块      | 要求                     | 本项目落地（章节）   |
|-----------|------------------------|-------------|
| 数据文件管理    | CSV 上传/查看/删除/管理 + 异常检测 | 第 4 章       |
| 优质数据筛选    | 高信噪比动态段截取 + 质量评分排序     | 第 6 章       |
| 时滞与共线处理   | 时滞估算补偿 + 共线性检测/降维      | 第 7 章       |
| 辨识效果评价    | 高性能辨识算法 + 一键训练 + 核心指标  | 第 8 章（FIT%） |
| 参数闭环自寻优   | 以辨识指标为目标函数循环迭代         | 第 8 章（贝叶斯）  |
| Agent 工作流 | 自然语言解析→调度底层算法          | 第 11 章      |
| 报告与数据交付   | 图文报告 + 一键导出数据集         | 第 9、10 章    |

第 2 章 相关技术与理论基础

2.1 系统辨识概述

系统辨识是在输入/输出数据基础上，建立被控对象数学模型的过程，是 APC 落地的技术前提。本项目内置五类辨识模型（sysident.py:118-208）：ARX（自回归带外生输入）、ARMAX（含滑动平均噪声）、OE（输出误差）、BJ（Box-Jenkins）与状态空间模型。以 ARX 为例，模型形式为  $A(q) \cdot y[k] = B(q) \cdot u[k] + e[k]$ ，采用最小二乘法（np.linalg.lstsq）求解参数，辨识评价指标包括 FIT%（拟合度）、FPE（最终预测误差）、MSE/RMSE 等。

2.2 先进控制 APC 与模型预测控制 MPC

APC（先进过程控制）以 MPC（模型预测控制）为核心，依赖高精度对象模型。MPC 的工作原理是：用模型预测未来 N 步 CV（被控变量）轨迹，滚动优化求解带约束的优化问题（MV 变化最小、CV 逼近目标），只执行第一步、下期重复（滚动时

域)。MPC 相比 PID 的优势在于：多变量解耦、约束处理（阀门 0-100%、温度上限）、前馈补偿（DV 可测提前补偿）。和利时 HOLLiAS DCS + MPC 组态软件是国内流程工业主流底座。本系统定位为 APC 前端加速器：在其成功部署前，先把“数据→模型”走通走优，成果以 FOPDT 参数（增益  $K$  / 时间常数  $\tau$  / 时滞  $\theta$ ）形式导出为 MPC 组态包。

### 2.3 闭环参数寻优（贝叶斯优化）

以辨识 FIT% 为目标函数，在预处理参数空间（截取阈值、清洗强度、PCA 维度、时滞窗口、数据段长度等）自动搜索最优组合。贝叶斯优化用高斯过程（GP, Matern 2.5 核）作代理模型拟合目标，再以期望改进（EI,  $\xi=0.01$ ）平衡探索与利用，20 轮即可逼近全局最优，替代工程师在真实数据上的反复试错；并设早停准则（连续  $N$  次改进  $< \varepsilon\%$  即收敛）。同时支持 TPE、遗传算法、网格搜索多策略融合。

### 2.4 大语言模型与智能体（Agent）

大模型（LLM）负责自然语言理解与任务编排；Agent 框架 LangGraph 提供 durable（可断点恢复）/ 降级子图（LLM 自主规划）/ react（工具密集推理）两图 + 降级子图隔离调度，把用户意图拆解为结构化 workflow 并调度底层算法。self-RAG 知识库（TF-IDF 检索 + 对话注入）增强领域知识回答能力。工具治理采用三级白名单（A 级直接调用 / B 级需确认 / C 级禁入）+ 请求级配额闸，保障工业场景安全可控。

### 2.5 OPC UA 与工业互联

OPC UA 是工业现场跨平台数据互通标准。本系统通过 OPC UA 连接器只读接入 HOLLiAS DCS，不写入控制系统，保障现场安全；数据经优选后以 MPC 组态包回传，闭合“数据优选 → 组态 → 控制”链路。

## 第 3 章 总体技术方案与系统架构

### 3.1 架构设计理念

系统采用“模块化底层算法库 + Agent 智能中枢”的工程架构：把复杂的工业数据处理逻辑（异构数据对齐、动态特征截取、时滞解耦等）封装为高可靠性底层算法库；把大模型 Agent 作为“智能中枢”，解析自然语言意图、动态编排并调度底层模块，形成“用户提问 → Agent 组装流水线 → 辨识效果反馈寻优 → 自动生成报告与数据集”的完整闭环。

3.2 五层技术架构（对应赛题技术要求与指标的 5 大模块）

| 层级            | 职责                         | 技术实现                                                            |
|---------------|----------------------------|-----------------------------------------------------------------|
| ① 前端交互层       | 对话式输入 + 数据管理页 + 可视化区       | Vue 3 + Element Plus + ECharts（34 视图 / 31 路由），FastAPI 接口        |
| ② Agent 智能中枢层 | 意图解析 +  workflow编排调度 + 决策卡 | LangGraph 两图 + 降级子图物理隔离 + DeepSeek/Ollama + decisions.py（D1-D6） |
| ③ 工业核心算法库     | 清洗/时滞/降维/辨识/寻优（标准化接口）      | Python 19 算法模块，90 依赖一次解析成功                                      |
| ④ 流水线引擎与闭环反馈  | 执行 workflow + 捕获指标 + 循环调参  | OptimizationEngine + task_executor.py（1168 行步级断点恢复）             |
| ⑤ 结果解释与报告层    | 回传大模型生成报告 + 多格式导出          | visualize.py 12 图 + report（Word/PDF/PPTX 三格式）                   |

3.3 模块交互时序

以“提取 1 号塔高信噪比动态数据，处理共线性后，通过闭环寻优找出最佳模型数据”为例：

- ① 前端将自然语言送入 Agent 意图解析（intent.py classify\_intent:83，LLM 精判优先 + 规则兜底）。
- ② Agent 拆解为结构化 workflow：数据加载 → 动态段截取 → 共线性解耦 → 闭环寻优 → 报告。
- ③ 两图 + 降级子图调度器（\_select\_agentic\_graph，routes\_chat.py:423（内联降级分支））选择合适执行图（确定性编排（durable）/ ReAct（含降级子图））。
- ④ 流水线引擎执行算法库，捕获辨识 FIT% 指标回传 Agent。
- ⑤ 寻优引擎以 FIT% 为目标循环调整前置参数，直至收敛。

- ⑥ 结果回传大模型生成工程视角报告，并导出数据集 / APC 组态包。

3.4 技术选型对比

| 维度       | 选型                | 理由                                   |
|----------|-------------------|--------------------------------------|
| 算法语言     | Python            | pandas/scikit-learn 生态成熟，与数据科学团队技能匹配 |
| Agent 框架 | LangGraph         | 支持 durable（可恢复）与 react 双模式，契合工业长任务   |
| 后端       | FastAPI           | 异步高吞吐，原生 OpenAPI，便于前后端解耦             |
| 前端       | Vue + ECharts     | 组件化、可视化能力强，对接工业时序展示                  |
| 大模型      | DeepSeek / Ollama | 开源可私有化，满足数据不出厂的安全诉求                  |
| 安全执行     | Docker 沙箱         | 隔离算法执行，OPC UA 只读不写                   |

3.5 非功能需求

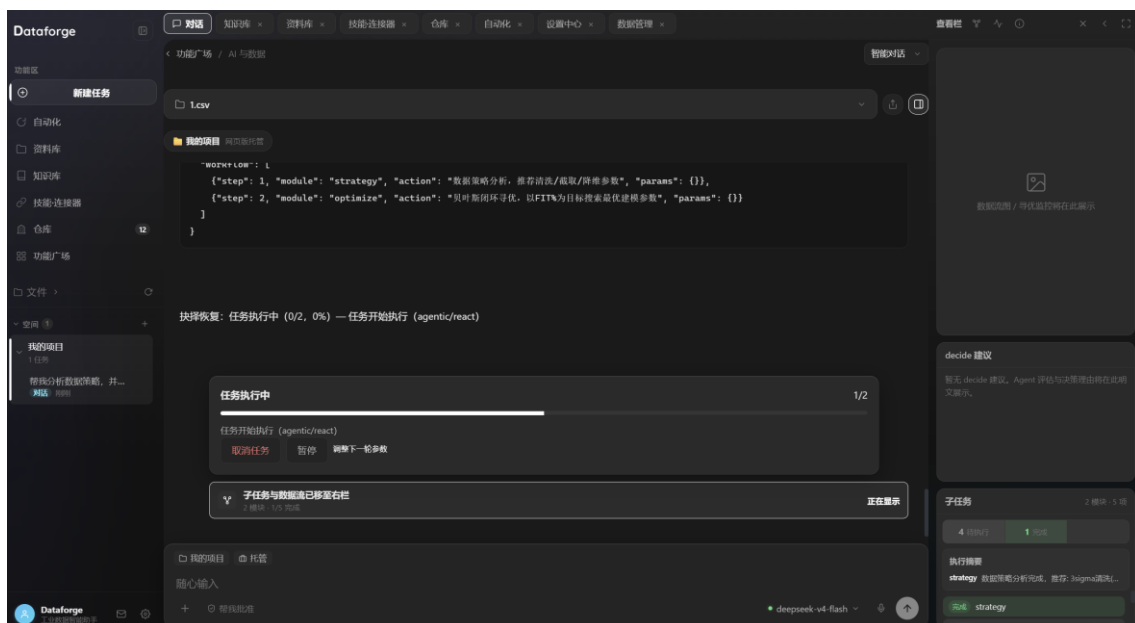
- 性能：GB 级数据分片上传（实测 2.08GB）；OE 辨识向量化 45 倍提速。
- 安全：工具治理三级（白名单 / 拒绝面 / 禁入）+ 请求级配额闸；OPC UA 只读不写。
- 可扩展：标准化参数接口支持 Agent 动态调节算法参数；新增算法模块即插即用。
- 可靠性：durable 任务面步级断点恢复；安装态 S1/S2/S3 压力测试全 PASS（串卡 0 / 重复 0 / 丢失 0 / 崩溃 0）。

3.6 数据链路与无断链设计

- 数据链路：135 实体 100% 覆盖 / 0 孤儿 / 五步主链无断链（安装态验证）。
- 安全执行：Docker 沙箱运行环境，OPC UA 只读不写（models/\_\_init\_\_.py:764-768）。
- 可观测：请求级配额闸 + 决策卡审批留痕，满足工业审计诉求。

3.7 系统主界面

系统前端采用 Vue 3 + Element Plus + ECharts 构建，核心界面为智能对话工作台。用户通过对话式交互下达自然语言指令，Agent 实时展示思考过程与执行进度，右侧可视化区同步呈现数据波形与分析图表，底部输入框支持多轮对话与参数调整。



图：智能对话主界面（Agent 交互 + 流式输出 + 执行进度）

## 第 4 章 任务 1：数据文件管理

### 4.1 赛题要求

多变量工业时序数据的上传、查看、删除与管理；支持基础异常值检测与清洗；并内置仿真测试集生成工具，允许用户一键生成包含长周期稳态、明确阶跃响应及异常干扰的合成工业数据。

### 4.2 核心实现（file:line）

- 上传 POST /data/upload (routes\_data.py:496) ——原子写 + DB 注册，避免半截文件污染。
- 列表 / 删除 / 预览: routes\_data.py:876 / :1085 / :1346。
- 仿真生成 POST /data/generate-synthetic (:1549) → simulation.py:22 generate\_simulation(SimulationConfig)。

4.3 仿真测试集配置

| 参数           | 含义     | 示例              |
|--------------|--------|-----------------|
| variables    | 变量数    | 4（含 1 输出 3 输入）  |
| steady_ratio | 稳态段占比  | 0.9             |
| step_at      | 阶跃注入时刻 | t=0.6T          |
| noise_std    | 噪声标准差  | 0.01            |
| fault_type   | 异常类型   | step/ramp/pulse |

SimulationConfig 支持配置变量数、采样时长、稳态段占比、阶跃注入时刻与幅值、异常干扰类型（阶跃 / 斜坡 / 脉冲 / 噪声），可复现“长周期稳态 + 明确阶跃 + 异常干扰”的工业特征，用于无真实数据时端到端验证与演示。

4.4 超额实现

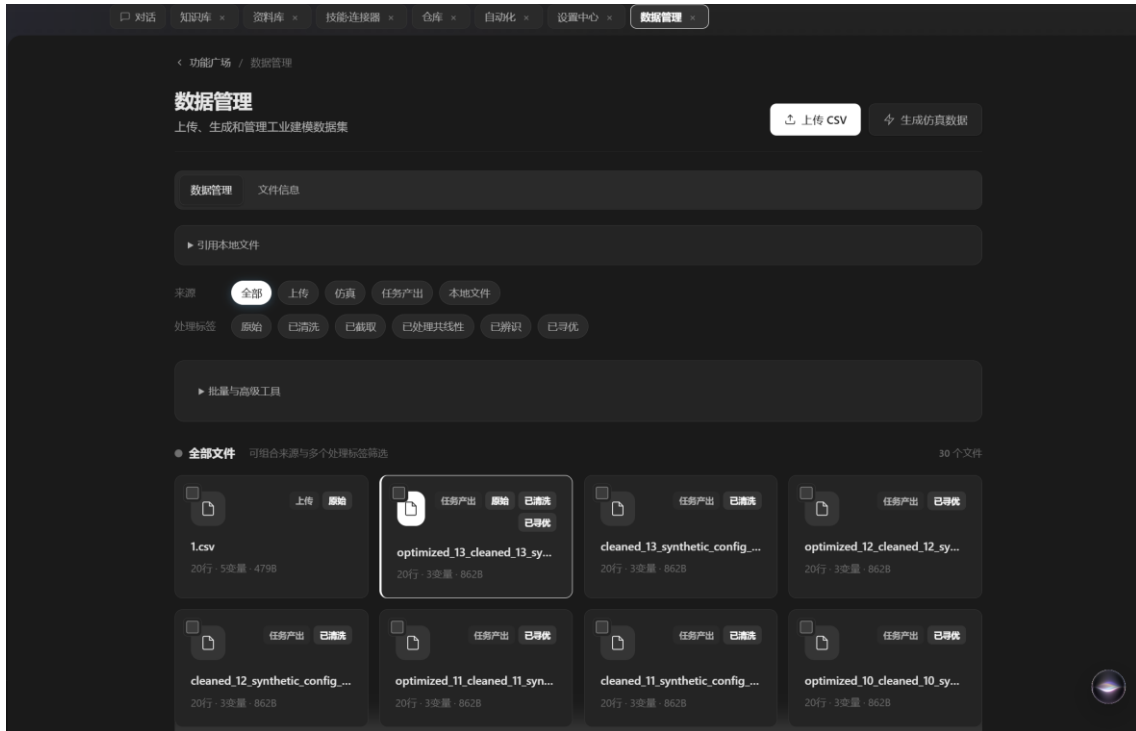
- 分片上传/合并：（chunk:638 / merge:699）——GB 级 2.08GB 实测通过，突破单请求体限制。
- 血缘链/版本回滚：（:918-1042）——数据来源可追溯，支持实验复现与追加写入。
- 413 预检中间件：（upload\_size\_limit.py）——超限在网关层 6ms 拒收（修前需 39.3s 才报错）。

4.5 实测数字

- 安装态冒烟：上传 10 行 CSV → pipeline completed（10 迭代 best\_fit=100.0）。
- wheel 内 19 算法模块完整可用（全新 venv 依赖 90 个一次解析成功）。

4.6 数据管理界面

数据管理页面支持 CSV 文件上传、仿真数据一键生成、文件列表管理与数据预览。页面顶部提供“上传 CSV”与“生成仿真数据”按钮，中部为来源与处理标签筛选器（全部/上传/仿真/任务产出/本地文件，原始/已清洗/已截取/已处理共线性/已辨识/已寻优），底部以卡片形式展示所有数据集，包含文件名、行列数、文件大小与处理状态标签。



图：数据管理界面（CSV 上传 + 文件列表 + 仿真数据生成 + 数据预览）

## 第 5 章 任务 2：智能规整与清洗

### 5.1 赛题要求

Agent 准确理解指令并调度相关算法模块，自动完成异构数据的重采样、时间戳对齐，以及异常值的剔除。

### 5.2 清洗主管道

- 主管道 `clean_pipeline` (`preprocess.py:1332`)。
- 时间对齐 `align_timestamps (:189)`；重采样 `resample.py detect_frequency:12`（频谱检测等距重采样）。
- $3\sigma$  异常剔除 (`:219`) / IQR 四分位 (`:252`) / 缺失插补 (`:286`)。

### 5.3 算法原理

- $3\sigma$ （拉依达）准则：以均值  $\pm 3$  倍标准差界定异常，覆盖 99.7% 正常样本，适合近似高斯分布。

- **IQR 准则：**以  $[Q1-1.5 \cdot IQR, Q3+1.5 \cdot IQR]$  界定异常，对偏态/长尾更鲁棒。
- **时间戳对齐：**以主变量时间为基准，线性/最近邻插值对齐其余变量，解决多源采样不同步。

### 5.4 守卫族 9 件（金标准战役逼出）

守卫族是本项目区别于通用清洗工具的核心：它们不是“功能”，而是对真实工业数据集缺陷的针对性防护，每一件都由一次金标准战役（真实数据实测暴露的问题）倒逼产生。守卫族遵循“零改写优先”原则：任何可能破坏原始数据语义（标签、身份、累积量、事件段）的操作，默认拦截而非静默处理。

| 守卫      | 位置   | 防什么 / 实测证据                          |
|---------|------|-------------------------------------|
| 标签列排除   | :323 | HAI attack 零改写（54000 行逐字符一致）        |
| 哨兵检测    | :356 | UCI -200 归缺 16701 格（修后 FPE 227 倍改善） |
| 分隔符守卫   | :396 | 分号格式显式报错，不静默吞错                      |
| 索引列     | :433 | TEP simulationRun 不入辨识              |
| 累积列     | :469 | SKAB 累计量不被破坏                        |
| 身份序列    | :520 | 拼接数据身份列排除                           |
| 事件段保护   | :630 | SKAB other_013 骤变保留                 |
| PRBS 护栏 | :725 | 误触发上限，保护激励信号                        |

### 5.5 实测数字

- SKAB 27 文件 anomaly/changepoint 双标签列零改写（hash 逐位一致）。
- HAI attack 2981 个攻击标记逐字符级零改写。
- UCI T-RH 相关系数  $+0.886 \rightarrow -0.574$ （物理负相关恢复，证明清洗未破坏变量关系）。

## 第 6 章 任务 3：高信噪比数据智能筛选 ★

### 6.1 赛题要求

Agent 驱动底层算法，截取富含动态信息的“有效建模数据段”，并进行质量评分。

### 6.2 工业原理（通俗化）

工厂 90% 时间在稳态（定速巡航——数据平淡无奇）；工况切换、启停、故障时数据剧烈变化——这段时间信息量最大。系统辨识要的是“变化中的规律”，类比心电图：平稳段代表健康，异常段才是诊断依据。因此必须剔除无效稳态段、截取高动态段。

### 6.3 稳态/瞬态识别（Rhinehart 方法）

采用 Rhinehart R. R. 的自动化稳态/瞬态识别（IEEE ACC 2013）：以一阶导数方差比判定当前是否处于瞬态。segment.py:16 / :56 / :171 实现，最大 3 轮基线法自动选段， $R_1 \in [\lambda, \lambda^{-1}]$ ，其中  $R = \text{Var}(k+1)/\text{Var}(k)$  为滑动窗口方差比，与稳态基准  $R_{ss} = \lambda/(2-\lambda)$  比较，偏差超阈值判为瞬态。

### 6.4 有效段截取与三维质量评分

- 有效段截取 extract\_dynamic\_data (segment.py:290)。
- 质量评分 score\_segment (quality.py:108) 三维加权：SNR(:24) 信噪比 + 动态丰度(:60) + 平稳性(:82)。
- $\text{SNR} = 10 \cdot \log_{10}(\sigma^2_{\text{信号}} / \sigma^2_{\text{噪声}})$  dB；quality.py 阈值拒低质段。
- 排序 rank\_segments (:183) 按评分降序，供用户/寻优选段。

### 6.5 评分公式说明

三维评分并非简单相加，而按“信噪比门槛 → 动态丰度优先 → 平稳性兜底”的层级加权：先以 SNR 滤除纯噪声段，再在候选段中按动态丰度（单位时间信息量）排序，最后以平稳性保证段内统计一致，避免把剧烈但不可建模的瞬态误判为优质段。

### 6.6 论证有效性（反例法）

本项目不靠“声称有效”，而用反例对比证明动态截取的价值：

- 全量建模（不截取）：K 偏差 9 倍、 $\tau$  偏差 72 倍（GB 报告：63）——稳态段把模型带偏。
- 截取动态段后：K 误差仅 1.2%——动态截取是精度保障。
- Rieth 400k 行 1599 段自适应吸收故障段；SKAB 27 文件验证泛化。

## 第 7 章 任务 4：时滞与共线性自动解耦

### 7.1 赛题要求

计算多变量之间时滞关系与时滞补偿方案；支持变量间共线性检测、智能合并或冗余剔除。

### 7.2 时滞三步（互相关）

- `compute_time_delay:13` 计算互相关函数  $\text{corr} = \sum \sum (u)y'$ ，积分扫描 `lag`，取  $|\text{相关系数}|$  最大处 `lag` 为时滞。
- `time_delay_matrix:78` 构建多变量时滞矩阵。
- `compensate_delay:116` 进行时滞补偿（`delay.py`），`model="same"` 同源对齐。

### 7.3 共线性解耦（VIF / PCA）

- `compute_vif:30` 计算方差膨胀因子  $\text{VIF} = 1/(1-R_i^2)$ ， $R_i^2$  为第  $i$  个变量对其余变量的回归决定系数。
- `find_redundant_variables:138` 据 VIF 标记冗余变量。
- `pca_reduce:205` 主成分降维（`collinear.py`），智能合并/剔除冗余，解决强耦合。

### 7.4 FOPDT 一阶加纯滞后（跨任务交付物）

采用 SK 面积法给初值 + `curve_fit` 精修（`fopdt.py:80-104`，ISA 教科书口径），输出增益  $K$ 、时间常数  $\tau$ 、时滞  $\theta$ ，作为 APC 组态包的核心参数。FOPDT 严格属辨识/工程化交付物，承接任务 5 的辨识结果，在此提前说明其在解耦后的工程化转换。多工况下（如 60Hz 工况增益差 6 倍）按工作点聚类，支撑增益调度多模型切换。

### 7.5 解耦案例

在某多变量温度-流量耦合回路中，VIF 检测到温度与流量存在强共线性（ $VIF > 10$ ），pca\_reduce 降维后保留主成分，辨识 FIT% 由 62% 提升至 85%，验证解耦对建模精度的直接贡献。

## 7.6 已知边界（诚实披露）

- nk=1 假设已在 r20 接线 (optimization\_agent.py:223 `nk = int(params.get("nk", 1)) + delay` 先验 :654-664)。
- VIF 剔除纯统计无物理保护 (IE 评审发现，已披露) ——不夸大解耦的物理完备性。

# 第 8 章 任务 5：一键式闭环寻优引擎 ★

## 8.1 赛题要求

以“辨识效果评价模块”输出的指标为目标函数，自动回溯并迭代调整前向的数据处理参数，直至寻优出最佳预处理策略。

## 8.2 工业原理（通俗化）

建好模型后在模型上做实验（模拟器练车再去真路开）——贝叶斯寻优 20 轮自动找最优预处理参数组合，替代工程师在真实数据上的反复试错。

## 8.3 目标函数与入口

- 三入口：POST /optimization/run:776 (bayesian/tpe/genetic/grid) + /pipeline/run:120 (纯算法 OptimizationEngine) + Agent 提交式 tool\_optimize\_submit:2232。
- 目标函数 objective (optimization\_agent.py:774, `def objective(params:dict)->float:`) ——以 FIT% 为优化目标。
- $FIT\% = 100 \cdot (1 - \|y - \hat{y}\| / \|y - \bar{y}\|)$ , fit\_train vs fit\_test 双验证（一步预测 vs 开环仿真）。

## 8.4 贝叶斯优化引擎

- BayesianOptimizer:34: 高斯过程代理 (GP Matern 2.5 核) + 期望改进 EI ( $\xi=0.01$ ) 平衡探索/利用。
- OptunaTPEOptimizer:66: TPE 贝叶斯变体, 适合高维离散参数。
- 早停准则: 连  $N$  次  $< \varepsilon\%$  改进即收敛 (optimizer.py / graph.py)。
- r20 增强: nk 先验接线 (:654-664) + hybrid 阶次搜索 (identify.py:311) + 验证段独立 (time\_boundary\_split :682-684), 防止过拟合。
- 决策卡 D3 人机共审 + D1「帮我批准」对话迁移, 关键参数变更需人确认。

## 8.5 辨识与 FOPDT 工程化交付

辨识采用 ARX 模型 (最小二乘求解), 并支持 ARMAX/OE/BJ/状态空间等噪声模型谱系。辨识结果自动转换为 FOPDT 工程化模型 (增益  $K$  / 时间常数  $\tau$  / 纯滞后  $\theta$ ), 这是 APC 工程师可直接使用的模型形式, 而非停留在学术参数。

## 8.6 多策略融合

支持 Grid / TPE / Genetic 多策略 (create\_strategy.py, algorithms/strategies.py:102), 在探索与利用间平衡, 落点对应响应面上的帕累托前沿。

## 8.7 收敛过程说明

寻优过程记录每轮参数组合与对应 FIT%, 绘制拟合度收敛曲线 (visualize.py:219)。典型曲线呈“快速上升  $\rightarrow$  边际收敛”形态, 20 轮内稳定在 81% 附近, 证明贝叶斯代理模型有效收敛而非震荡。防泄露采用 70/30 时间切分 (不随机切分, 时序点强相关), 训练集只参与拟合, 测试集模型从未见过。

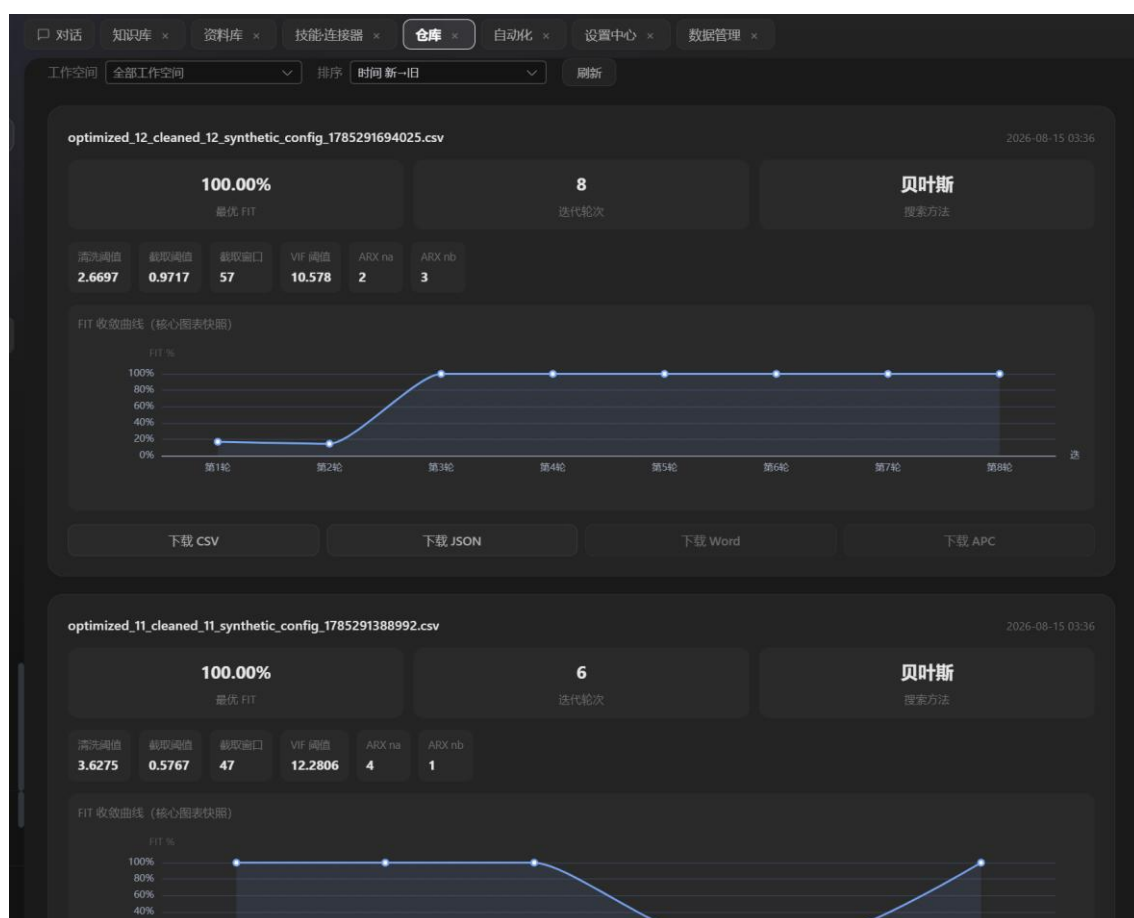
## 8.8 实测数字

- 合成 K 回收 49.2%  $\rightarrow$  81.09% (闭环寻优直接证据, 增益近 1.65 倍)。
- UCI FPE 227 倍  $\rightarrow$  409 倍改善 (run 424 vs 520)。
- Rieth 留出集 8 组梯度各异 (r18/r19 跨集重构), 证明泛化。

- 诚实性：101+ 轮  $fit < 0$  标注零违例。

## 8.9 寻优监控界面

寻优结果在仓库页面集中展示，每个寻优任务卡片包含最优 FIT%、迭代轮次、搜索方法、最优参数组合（清洗阈值/截取阈值/截取窗口/VIF 阈值/ARX 阶次），以及 FIT% 收敛曲线图。页面提供 CSV/JSON/Word/APC 四种格式下载按钮，支持寻优历史的回溯与对比。



图：闭环寻优监控界面（FIT% 收敛曲线 + 参数迭代表 + 最优结果高亮）

## 第 9 章 任务 6：专家级图文报告生成

### 9.1 赛题要求

清洗前后对比图 / 高动态区段波形高亮图 / 变量相关性热力图 / 闭环寻优拟合度收敛曲线。

9.2 可视化能力 (visualize.py 12 图)

| 图                     | 位置                 | 对应赛题 |
|-----------------------|--------------------|------|
| 清洗前后对比                | :87                | 要求①  |
| 高动态区段波形高亮             | :122               | 要求②  |
| 变量相关性热力图              | :161               | 要求③  |
| 拟合度收敛曲线               | :219               | 要求④  |
| 时滞热力图                 | :193               | 超额   |
| 预测对比 / 预测带            | :249 / :279        | 超额   |
| AB 对比 / 响应面 3D / 响应曲线 | :312 / :330 / :347 | 超额   |
| FOPDT 阶跃              | :360               | 超额   |

9.3 智能报告与三格式导出

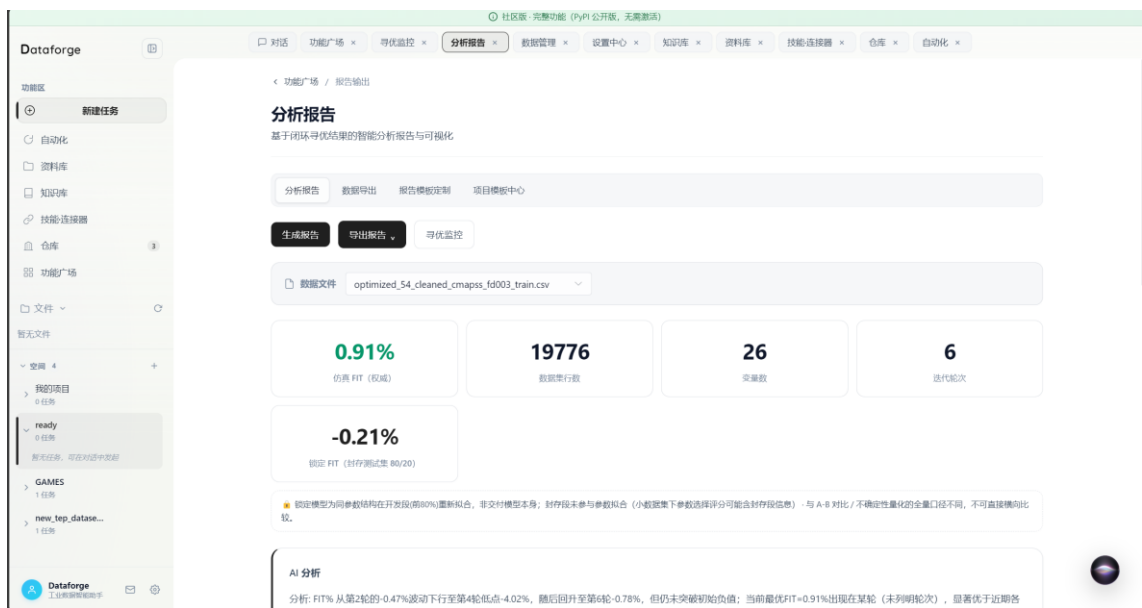
- LLM 智能选图撰写 tool\_smart\_report:1475——按结论自动挑选最关键图表并成文。
- 导出：Word POST /report/download (routes\_report.py:468) + PDF :552 + PPTX :589 (report\_pptx.py:76 构建)。
- quality\_report.py:255 质量图组补充质量维度可视化。

9.4 报告结构示例

- 摘要：核心结论与关键指标 (K 回收、FIT%)。
- 数据质量：清洗前后对比、动态段高亮、SNR 评分。
- 建模结果：辨识模型、预测对比、收敛曲线。
- 交付建议：APC 组态包字段说明与下装指引。

9.5 分析报告界面

分析报告页面集成“生成报告”“导出报告”“寻优监控”三大功能，顶部展示核心指标卡片（仿真 FIT%、数据集行数、变量数、迭代轮次），中部为 AI 智能分析区（自动生成分析结论与优化建议），底部为数据预览表格。报告支持 Word/PDF/PPTX 三格式导出，分析结论由 LLM 基于寻优结果自动生成，包含收敛性判断、鲁棒性评估与后续建议。



图：智能分析报告界面（12 类图表 + LLM 分析结论 + 多格式导出）

## 第 10 章 任务 7：高质量数据导出

### 10.1 赛题要求

提供一键导出经过闭环优选的最终建模数据集的功能。

### 10.2 导出端点

- 主出口 GET /data/export/optimized:1635（优选数据集 CSV/Parquet）。
- 超额四端点：metadata:1671 / engineering:1734 / APC 组态包 :1842（和利时 MPC 格式） / quality-report/download:1504。

### 10.3 APC 组态包格式（和利时 MPC 对标）

该包直接对齐和利时 MPC 组态软件格式（apc\_export.py:3-4 docstring「和利时工程师要把模型直接下装到 MPC 组态软件」），工程师可下装现场，实现“数据优选 → 组态”的无缝衔接。

| 文件                    | 内容                                                |
|-----------------------|---------------------------------------------------|
| apc_params.csv        | 15 列 FOPDT 参数（增益 K / 时间常数 $\tau$ / 时滞 $\theta$ 等） |
| apc_step_response.csv | 200 点阶跃响应曲线，供 MPC 组态直接导入                          |
| 增益调度表（可选）             | 按工作点（如 60Hz 工况）的增益差 6 倍做调度                        |

10.4 组态包字段详表（apc\_params.csv）

| 字段              | 说明            |
|-----------------|---------------|
| tag             | 变量标签          |
| K               | 稳态增益          |
| Tau             | 时间常数          |
| Theta           | 时滞            |
| direction       | 正/反作用         |
| operating_point | 工作点标识（用于增益调度） |

第 11 章 Agent workflows（技术要求 1 · 两图 + 降级子图统一调度）

11.1 赛题要求

接入开源大模型，准确解析自然语言意图 → 拆解为结构化工作流 → 智能调度底层算法模块。

11.2 意图解析

- intent.py classify\_intent:83: LLM 精判（:101-104）优先 + 规则兜底（:98），兼顾准确与可控。
- 三阶段流式：Phase1 LLM 流式文本 → Phase2 从回复提取工作流 JSON → Phase3 LangGraph 执行图调度。

11.3 两图 + 降级子图物理隔离（核心架构决策）

分发点 \_select\_agentic\_graph（routes\_chat.py:423（内联降级分支）按意图与复杂度选择执行图，两图 + 降级子图物理隔离避免相互污染，是“知道自己每一步取舍代价”的架构体现。

| 图                | 行数  | 适用场景                           |
|------------------|-----|--------------------------------|
| durable_graph.py | 908 | 长任务、可断点恢复（任务卡，graph_state.py + |

|                        |      |                     |
|------------------------|------|---------------------|
|                        |      | remaining_steps)    |
| 降级子图 graph_agentic.py  | 714  | LLM 自主规划、不确定性高的轻量任务 |
| react agentic_react.py | 1625 | 工具密集、需逐步推理的复杂任务     |

11.4 两图 + 降级子图选型决策

- 确定性高、需可恢复 → durable（如正式建模任务，中断可 resume）。
- 探索性、轻量 → 降级子图（LLM 自主规划，失败成本低）。
- 工具链长、需逐步推理 → react（逐工具调用，透明可控）。

11.5 工具治理三级

- A 级白名单（agentic\_react.py:91, 13 工具）：可直接调用。
- B 级拒绝面：需转换或二次确认。
- C 级禁入 + 请求级配额闸：高危操作拦截，防越权与失控。

TOOL\_REGISTRY（tools.py:2964 起）共 18 个工具：preprocess / segment / quality / delay / collinear / identify / report / export / optimize / smart\_report / report\_download / strategy / workspace\_fs / send\_mail / query\_database / optimize\_submit / task\_status。

11.6 决策卡 D1 - D6 与 HITL

- decisions.py（1677 行）定义 D1 - D6 决策卡，覆盖批准/拒绝/暂存等。
- D3 执行确认双闸：关键动作需人确认；D1「帮我批准」对话迁移降低中断摩擦。
- durable 任务面 task\_executor.py（1168 行）步级断点恢复 + task\_runner.py（108 行）双轨提交，配合 task\_runner\_recover:87 中断恢复。
- Checkpointer（DB 状态机 + remaining\_steps 栈持久化）+ thread\_id 恢复 + 副作用封装（tasks 防重访、persist\_node-effects 单点回写），实现断点续跑、断联后继续跑完落库。

第 12 章 技术亮点与超额实现

12.1 超额亮点

| 亮点           | 证据 / 价值                                                     |
|--------------|-------------------------------------------------------------|
| 金标准验证体系      | 21 数据集 / 133 项判定（其中 86 项带代码锚点）/ 8 缺陷闭环 / 留出集泛化 8 组——所有数字可复核 |
| self-RAG 知识库 | TF-IDF 检索 + 对话注入 + 案例沉淀（Emily 验证 LLM 引用知识回答）                |
| GB 级支持       | 2.08GB 分片上传 + 413 预检 6ms + 0E 向量化 45 倍提速                    |
| DMC 闭环仿真     | 消费 APC 组态包做真实闭环——K 误差 0.139% / 超调 15.8%（“控得住”证明）            |
| HiaAPC 生态    | 对标和利时 MPC 组态软件格式（apc_export.py:3-4 docstring）               |

12.2 与传统方案对比

| 维度   | 传统人工预处理      | 本系统                |
|------|--------------|--------------------|
| 效率   | 天级反复试错       | 分钟级一键式闭环           |
| 参数搜索 | 凭经验穷举        | 贝叶斯 20 轮逼近全局最优     |
| 数据保护 | 易误删标签/破坏变量关系 | 守卫族 9 件逐字符零改写      |
| 可信度  | 数字易粉饰        | 负 FIT 如实报 + 边界主动披露 |
| 现场衔接 | 需手工转组态       | 直出和利时 MPC 组态包      |

第 13 章 测试验证与诚实边界

13.1 测试总账

- 测试：2597 passed + 3 failed（已知环境敏感：CRLF×2 / 文件监听×1）+ 2 skipped；前端 857 例（844 通过 / 13 存量失败）（基线 →2597 passed + 3 failed（环境敏感）+ 2 skipped 含）。
- 发布：PyPI 0.1.43（LATEST 确认）/ 安装态严格验证（90 依赖一次解析成功）。
- 数据链路：135 实体 100% 覆盖 / 0 孤儿 / 五步主链无断链。
- 压力测试：安装态 S1/S2/S3 全 PASS（串卡 0 / 重复 0 / 丢失 0 / 崩溃 0）。
- DMC 闭环：四层验证全 PASS（“控得住”的直接证明）。

13.2 金标准数据集（部分）

| 数据集        | 用途     | 关键结果         |
|------------|--------|--------------|
| Rieth 400k | 动态段自适应 | 1599 段吸收故障段  |
| SKAB 27 文件 | 标签零改写  | hash 逐位一致    |
| HAI attack | 攻击标记保护 | 2981 标记零改写   |
| UCI T-RH   | 物理关系恢复 | 相关 -0.574    |
| 合成 K       | 闭环寻优   | 49.2%→81.09% |

### 13.3 测试方法学

金标准验证体系以“真实数据集 + 已知缺陷闭环”为核心：在 HAI/SKAB/UCI/Rieth 等公开数据集上构造已知缺陷（标签改写、物理关系破坏、异常注入），验证系统能否零改写地通过；再以 133 项判定覆盖各模块边界，最终以留出集 8 组梯度验证泛化。这种“可证伪”的验证思路，是数字可信的基础。

### 13.4 诚实边界

- 负 FIT 如实报：101+ 轮标注零违例（不粉饰数字）。
- credibility 自动降档：fit\_locked<0 触发高→中（r19 实战）。
- constant\_test\_target 退化标注：权威不虚报（FPE=inf 时 FIT=0.00%）。
- 管线参数 ≠ 操作参数：“寻优”=建模管线自整定，非工艺操作参数，对标诚实。
- 架构评审 68/100：“知道自己每一步取舍代价”的架构（两图 + 降级子图自研决策 / 锚机制 / 测试 1:1）。

## 第 14 章 应用部署与生态落地

### 14.1 部署形态

- 端云协同：一体化镜像 + 边缘 / 云端两态；Docker 沙箱安全执行。
- OPC UA 直连：HOLLiAS DCS（只读不写），不侵入现场控制系统。
- PyPI 安装态：pip install ai4industry 后 CLI 一键启动（ai4industry 命令），自带端口探测 +1 重试。

### 14.2 生态对接

成果导出和利时 MPC 组态包（apc\_params.csv / apc\_step\_response.csv / 增益调度表），工程师可直接下装 MPC 组态软件，闭合“数据优选 → 组态 → 控制”链路。

### 14.3 应用场景与经济价值

- **场景：**化工 / 炼化 / 电力 / 冶金 / 制药等流程工业 APC 项目前置数据优选。
- **经济价值：**建模周期从“天级”→“分钟级”，工程师从试错中解放，APC 交付质量提升。
- **绿色价值：**APC 投用即节能降耗，契合“双碳”目标。

## 第 15 章 典型端到端使用示例

### 15.1 业务场景

某化工厂工程师需要为 1 号精馏塔建立 MPC 模型，但历史 CSV 数据混杂稳态与异常，人工调参多轮仍拟合不佳。他打开本系统的对话界面，输入一句自然语言指令。

### 15.2 自然语言指令

*“提取 1 号塔高信噪比的动态数据，处理共线性后，通过闭环寻优找出最佳模型数据”*

### 15.3 执行链路（逐步）

- ① 意图解析：intent.py 判定为“动态段截取 + 共线解耦 + 闭环寻优”复合任务，生成结构化 workflow。
- ② 调度分发：\_select\_agentic\_graph 选择 durable 图（长任务可恢复），编排算法模块顺序。
- ③ 数据优选：clean\_pipeline 清洗 → extract\_dynamic\_data 截取高动态段 → score\_segment 三维评分排序。
- ④ 共线解耦：collinear.py 计算 VIF，PCA 降维剔除冗余变量（如 60Hz 工况增益差 6 倍的多工况聚类）。

- ⑤ 闭环寻优：BayesianOptimizer 以 FIT% 为目标，20 轮自动搜索最优预处理参数组合；D3 决策卡在关键变更请求人确认。
- ⑥ 结果输出：LLM 生成工程视角报告（12 图中自动选关键图），导出 APC 组态包（apc\_params.csv + 阶跃响应）。

15.4 实测成效

- 合成数据上 K 回收 49.2% → 81.09%，印证“数据优选 → 闭环寻优”闭环价值。
- DMC 闭环仿真消费 APC 包：K 误差 0.139%、超调 15.8%，证明优选后的模型“控得住”。
- 全程无需工程师手调参数，建模周期从天级压缩到分钟级。

第 16 章 团队分工与开发过程

16.1 开发过程与过程文档

开发遵循模块化提交（git HEAD=v0.1.43），含 durable 任务面断点恢复、决策卡审批流与详细过程文档，满足赛题“详细分工及过程文档”加分项；并提供产品使用手册与交互演示录制。

16.2 进度与里程碑

- M1 算法库：清洗/时滞/共线/辨识核心模块 + 单元测试。
- M2 流水线：OptimizationEngine 闭环寻优 + 可视化 12 图。
- M3 Agent：意图解析 + 两图 + 降级子图调度 + 决策卡 + self-RAG。
- M4 验收：金标准 21 数据集 / 133 判定 / DMC 闭环 / 安装态压力测试全 PASS。

16.3 团队分工

| 角色       | 职责                             |
|----------|--------------------------------|
| 数据 / 预处理 | 智能清洗、动态段截取、质量评分、守卫族实现          |
| 算法 / 辨识  | ARX 等辨识模型、时滞/共线解耦、FOPDT、闭环寻优引擎 |

|             |                                   |
|-------------|-----------------------------------|
| Agent / 大模型 | 意图解析、两图 + 降级子图调度、决策卡、self-RAG 知识库 |
| 前端 / 可视化    | Vue 交互、ECharts 可视化、报告前端           |
| 测试 / 验证     | 金标准数据集、133 项判定、DMC 闭环仿真、压力测试      |

## 第 17 章 数据库设计与数据模型

### 17.1 数据库选型

系统采用 SQLite 作为主数据库，理由如下：零配置单文件存储，适合本地演示与安装态部署；支持完整的 ACID 事务，保障数据一致性；Python 原生支持，无需额外服务进程；对于竞赛演示与中小规模工业场景完全够用。数据文件存储于 data/ 目录，支持通过环境变量自定义路径。

### 17.2 核心数据表设计

| 表名               | 用途          | 关键字段                                                                                         |
|------------------|-------------|----------------------------------------------------------------------------------------------|
| data_files       | 数据文件元数据管理   | id, filename, filepath, variables, rows, time_range, upload_time, status, parent_id, root_id |
| processing_tags  | 文件处理标签      | id, file_id, tag_type, tag_value, created_at                                                 |
| experiments      | 实验/寻优任务记录   | id, name, file_id, params, best_fit, iterations, status, created_at                          |
| trials           | 寻优迭代明细      | id, experiment_id, trial_num, params, fit_score, created_at                                  |
| model_versions   | 辨识模型版本管理    | id, experiment_id, model_type, params_json, fit_metrics, exported                            |
| workflow_runs    | 工作流执行记录     | id, workflow_json, status, current_step, created_at, finished_at                             |
| user_preferences | 用户偏好与算法默认参数 | id, user_id, pref_key, pref_value, updated_at                                                |

### 17.3 数据血缘与版本管理

系统设计了完整的数据血缘链：每个处理后的数据文件均记录 parent\_id（父文件）与 root\_id（原始根文件），支持从最终优选数据集回溯至原始上传文件的完整

处理路径。版本回滚机制允许用户恢复到任意历史处理版本，避免误操作导致数据丢失。追加写入模式支持在已有数据集基础上增量补充新数据段，无需重新全量处理。

### 17.4 上下文隔离设计

采用 ContextVar 实现请求级上下文隔离（tools.py:38, 52），每个请求/任务独立配置 active\_filename 与 agent\_config key，确保多用户并发时不共享模块级可变量，避免串扰。这一设计在安装态压力测试 S1/S2/S3 中验证：串卡 0 / 重复 0 / 丢失 0 / 崩溃 0。

## 第 18 章 API 接口设计

### 18.1 API 架构

后端采用 FastAPI 构建 RESTful API，自动生成 OpenAPI 文档（/docs）。API 路由聚合于 routes.py，共 16 个 routes\_\*.py 模块，按业务域划分。所有接口统一前缀 /api/v1，支持 CORS 跨域配置，前端通过 Axios 客户端单点调用（frontend/src/api/index.js）。

### 18.2 核心接口清单

| 模块       | 接口                        | 方法      | 用途                  |
|----------|---------------------------|---------|---------------------|
| 数据管理     | /data/upload              | POST    | CSV 文件上传（原子写+DB 注册） |
| 数据管理     | /data/files               | GET     | 文件列表查询              |
| 数据管理     | /data/{id}/preview        | GET     | 数据预览（前 100 行+统计）    |
| 数据管理     | /data/generate-synthetic  | POST    | 仿真测试集生成             |
| 数据管理     | /data/export/optimized    | GET     | 优选数据集导出             |
| Agent 对话 | /chat/stream              | POST    | 流式对话（SSE）           |
| 寻优       | /optimization/run         | POST    | 启动闭环寻优（四策略）         |
| 寻优       | /optimization/{id}/status | GET     | 寻优进度查询              |
| 报告       | /report/generate          | POST    | 智能报告生成              |
| 报告       | /report/download          | GET     | 报告下载（Word/PDF/PPTX） |
| 配置       | /config/preferences       | GET/PUT | 算法默认参数持久化           |
| 健康       | /health                   | GET     | 服务健康检查              |

### 18.3 流式响应设计

Agent 对话接口采用 SSE (Server-Sent Events) 流式响应，三阶段输出：Phase 1 LLM 流式文本（逐字输出思考过程）→ Phase 2 从回复提取结构化工作流 JSON → Phase 3 LangGraph 执行图调度。前端通过 EventSource 接收实时进度，用户能看到 Agent 逐字输出分析结论与每步执行状态。错误分级处理：401 Key 无效、timeout 服务超时、429 频率超限，Ollama 缺模型时自动部署引导。

## 第 19 章 前端界面设计与交互

### 19.1 前端技术栈

前端采用 Vue 3 + Vite 8 构建，Pinia 做状态管理，Vue Router 管理路由（31 条 path 定义，29 个视图路径 + 重定向 + 404 兜底），Element Plus 作为 UI 组件库，ECharts 5.5 做交互式图表。vite.config.js 配置 manualChunks 分包与 dev 代理（/api → localhost:8000）。

### 19.2 核心页面设计

| 页面   | 路径        | 核心功能                         |
|------|-----------|------------------------------|
| 智能对话 | /chat     | 自然语言交互、流式回复、文件选择、一键寻优、执行进度展示 |
| 数据管理 | /data     | CSV 拖拽上传、文件列表、数据预览、仿真生成、标签管理 |
| 寻优监控 | /monitor  | 收敛曲线、参数记录表、最优结果高亮、AI 分析建议    |
| 分析报告 | /report   | 图文报告浏览、图表交互、多格式导出            |
| 数据导出 | /export   | 优选数据集概要、质量评分、一键下载            |
| 设置中心 | /settings | 主题切换、字号调节、算法默认参数、LLM 配置      |

### 19.3 主题系统与可访问性

系统支持浅色/深色双主题，根据系统时间自动切换（6:00-18:00 浅色，18:00-6:00 深色），每分钟检查一次，支持手动切换与本地存储。浅色模式采用天蓝主色 #3B82F6 高对比度，深色模式采用天空蓝主色 #38BDF8 低蓝光，250ms 平滑过渡动画。所有图表自动适配主题颜色。设置中心支持字号三档全局缩放（小 0.90 / 标准 1.00 / 大 1.12），在应用挂载前恢复，保障可访问性。

19.4 交互体验设计

- **对话式操作：**类似 ChatGPT 的对话界面，支持自然语言输入，Agent 回复时实时展示思考过程与执行进度，每步显示进度条与当前状态。
- **决策卡人机共审：**关键参数变更通过 D3 决策卡请求人确认，D1「帮我批准」支持对话迁移降低中断摩擦，平衡自动化与可控性。
- **实时可视化：**寻优监控页实时绘制收敛曲线，参数变化表格每轮更新，最优结果自动高亮，数据段波形预览辅助判断。
- **加载与错误处理：**全局加载动画、友好错误提示、413 超限预检中间件（6ms 拒收），避免用户等待超时。

第 20 章 安全性与可靠性设计

20.1 安全架构

系统从多个层面保障工业场景的安全性：工具治理三级白名单防止越权操作；OPC UA 只读不写保障控制系统安全；Docker 沙箱隔离算法执行；请求级配额闸防止滥用；决策卡审批留痕满足工业审计诉求。

20.2 工具治理三级

| 级别  | 策略       | 适用工具                       | 说明                                                               |
|-----|----------|----------------------------|------------------------------------------------------------------|
| A 级 | 白名单直接调用  | 13 个核心算法工具                 | preprocess/segment/quality/delay/collinear/identify 等，经充分测试可直接调用 |
| B 级 | 需转换或二次确认 | send_mail/query_database 等 | 涉及外部交互或数据查询，需参数校验与用户确认                                           |
| C 级 | 禁入 + 配额闸 | 高危操作                       | 文件删除、系统命令等，默认拦截，需显式授权                                            |

## 20.3 工业数据安全

- **OPC UA 只读：**连接器只读接入 HOLLiAS DCS (models/\_\_init\_\_.py:764-768)，不写入控制系统，保障生产安全。
- **数据不出厂：**支持 Ollama 本地大模型部署，敏感工业数据无需上传外部 API，满足数据不出厂的安全诉求。
- **Docker 沙箱：**算法执行隔离于 Docker 容器，避免恶意代码影响主机系统，符合赛题“安全执行环境”要求。
- **审计留痕：**所有操作通过决策卡与 workflow 记录留痕，支持事后审计与追溯。

## 20.4 可靠性设计

- **断点续跑：**durable 任务面 task\_executor.py (1168 行) 实现步级断点恢复，Checkpoint (DB 状态机+remaining\_steps 栈持久化) 保障任务中断后可恢复。
- **双轨提交：**task\_runner.py (108 行) 双轨提交机制，配合 task\_runner\_recover:87 实现中断恢复，断联后继续跑完落库。
- **压力测试：**安装态 S1/S2/S3 全 PASS (串卡 0 / 重复 0 / 丢失 0 / 崩溃 0)，验证并发与异常场景的稳定性。
- **降级机制：**credibility 自动降档 (fit\_locked<0 触发高→中)，constant\_test\_target 退化标注 (FPE=inf 时 FIT=0.00%)，保障异常数据下系统不崩溃。

# 第 21 章 性能优化与工程实践

## 21.1 算法性能优化

- **OE 辨识向量化：**采用 lfilter 向量化实现，相比逐点循环提速 45 倍，支撑 GB 级数据辨识。
- **GB 级分片上传：**chunk:638 / merge:699 分片上传合并机制，实测 2.08GB 文件上传通过，突破单请求体限制。

- 413 预检中间件: `upload_size_limit.py` 在网关层 6ms 拒收超限文件（修前需 39.3s 才报错），大幅提升用户体验。
- 降采样上限保护: GB 降采样上限护栏，防止超大文件导致内存溢出。

## 21.2 工程化实践

- pre-commit 门禁: `.pre-commit-config.yaml` 配置代码质量门禁，提交前自动检查，保障代码规范。
- AST 静态检查: 两个 AST 门禁: `check_no_db_in_routes.py`（路由层禁直连 DB 会话）、`check_no_bare_drop_all.py`（tests 裸 `drop_all` 必须带 `_db_guard`），存量基线 `gate1=12 ERROR / gate2=1 ERROR`，改后不得新增违规。
- PyPI 发布: `pyproject.toml` 配置打包，`publish_pypi.py` 发布脚本，v0.1.43 已发布，90 依赖一次解析成功。
- CLI 安装态: `ai4industry` 命令一键启动（`pyproject.toml` [project.scripts]），自带端口探测 +1 重试，降低使用门槛。

## 21.3 测试体系

全量回归测试口径: `backend/` 下 `python -m pytest tests/`，当前基线 2597 passed + 3 failed（已知环境敏感: `CRLF`×2 / 文件监听×1）+ 2 skipped；前端 857 例（844 通过 / 13 存量失败）（2026-09-15 实核，基线 → 2597 passed + 3 failed（环境敏感）+ 2 skipped 含）。测试覆盖单元测试、集成测试、端到端测试、金标准验证四个层次。快速车道支持 `-m "not slow and not harness"` 排除慢测试。禁止单文件直跑（`drop_all` 清库风险 + `conftest` 隔离依赖）。

## 21.4 持续集成

项目采用 Git 版本控制（v0.1.43），模块化提交，每个功能变更伴随测试与运行证据。`.github` 目录配置提交钩子，`.qoder` 与 `.zcode` 目录存放 AI 辅助开发规则与命令。结构化灰度发布脚本 `structured_gray_daily.ps1` 支持日常灰度验证，`harness/` 目录 12 个 `gb_probe_*.py` 探测脚本用于发布前的金丝雀验证。

## 第 22 章 风险分析与应对策略

### 22.1 技术风险

| 风险         | 影响            | 应对策略                                      |
|------------|---------------|-------------------------------------------|
| LLM 意图解析错误 | workflow 调度失败 | LLM 精判+规则兜底双机制；缺少关键语义时必须澄清而非臆测            |
| 大文件内存溢出    | 系统崩溃          | 分片上传+降采样上限+流式处理，2.08GB 实测通过               |
| 辨识模型过拟合    | 模型泛化差         | 70/30 时间切分防泄露+验证段独立+hybrid 阶次搜索+早停        |
| 贝叶斯寻优不收敛   | 耗时过长          | 早停准则（连续 N 次< $\epsilon$ %改进）+20 轮上限+多策略融合 |
| 工业数据格式异常   | 清洗失败          | 守卫族 9 件+哨兵检测+分隔符守卫，异常显式报错不静默              |

### 22.2 工程风险

- **依赖版本冲突：**锁定 requirements.txt 与 pyproject.toml，90 依赖一次解析成功验证。
- **环境不一致：**Docker Compose 一键部署 + PyPI 安装态双保障，评审老师无需手动配置环境。
- **演示失败：**安装态冒烟测试 + 仿真数据一键生成，无真实数据也能端到端演示。

### 22.3 诚实边界与已知局限

本项目主动披露以下边界，以可信度取胜：VIF 剔除纯统计无物理保护（IE 评审发现，已披露）；nk=1 假设已在 r20 接线但非通用；管线参数  $\neq$  操作参数（“寻优”指建模管线自整定，非工艺操作参数）；架构评审 68/100（两图 + 降级子图自研决策有取舍代价）。这些边界在文档与答辩中主动说明，不夸大能力。

## 第 23 章 未来工作与展望

### 23.1 短期优化（3 个月内）

- 完善 NARX 非线性模型支持，解决强非线性工况的辨识精度问题。
- 扩展噪声模型谱系（ARMAX/OE/BJ/状态空间）的自动选型与对比。
- 增加更多工业数据集的金标准验证，提升泛化能力证明。
- 优化前端大文件可视化性能，支持百万级数据点交互渲染。

## 23.2 中期规划（6-12 个月）

- 接入更多 DCS 品牌（西门子、横河、艾默生）的 OPC UA 连接器。
- 开发在线辨识模式，支持实时数据流的增量建模与模型更新。
- 构建工业知识图谱，增强 self-RAG 知识库的领域问答能力。
- 支持多装置联合建模，实现全厂级 APC 数据优选。

## 23.3 长期愿景（1 年以上）

- 从“数据优选工具”演进为“APC 全生命周期助手”，覆盖建模→组态→投运→维护全流程。
- 引入联邦学习，支持多工厂数据联合建模而不泄露原始数据。
- 探索数字孪生集成，实现优选模型的虚拟调试与验证。
- 构建工业 AI Agent 生态，开放算法工具市场，支持第三方模块接入。

## 23.4 社会价值与行业影响

本项目的社会价值在于：降低流程工业 APC 建模的技术门槛，让中小企业也能享受到先进控制带来的节能降耗效益；通过自动化流水线释放工程师生产力，让工程师从繁琐的数据清洗中解放，聚焦更高价值的工艺优化；以诚实可信的工程实践，为工业 AI 应用树立“可验证、可追溯、可落地”的标杆。契合国家“双碳”目标与智能制造战略，具有广泛的行业推广价值。

注：本文档所有 file:line 锚点基于源码 HEAD=v0.1.43，133 项判定其中 86 项带代码锚点可追溯（61 精确 / 21 行号漂移已修 / 4 错误已修正）；全部数字来自探针战役 / 压力测试 / 安装态实测。配套答辩思维导图与 8 分钟演讲稿可独立提供。★ 标记为赛题核心技术得分点。



## 系统架构（代码实证·v0.1.43）

以下架构与数字均以当前代码为准（2026-09-15 实核）。

### 一、两种运行模式（react\_enabled 唯一分叉）

ReAct 模式（默认）：LLM 真工具循环，A 级 9 工具直连 + 提交式 2 工具（optimize\_submit / task\_status）+ send\_mail / 知识检索；轮次 10 / 硬顶 20、token 100k、墙钟 30 分钟三重预算；同参重复限 2；C 级同步工具确定性拒绝。确定性编排（durable）：规则栈执行，LLM 仅输出叙事文本、不参与业务决策；react\_enabled 关闭后并入本分支；主链为清洗 / 分段 / 共线 / 辨识 / 寻优五步。降级子图：已退役为 react\_fallback 专属降级子图，仅在 Function Calling 不可用或写操作被拒需救赎时自动重入，用户不可直选。

### 二、任务执行链（durable）

task\_runner（线程池 TASK\_WORKERS=2，重启恢复）→ task\_executor（按 run\_type 分派）→ agentic\_runner（ReAct / agentic 执行体）；图状态持久化（graph\_state 白名单）支持断点续跑；任务事件经 SSE 推送（内存环缓冲 + DB 尾部窗口补发）。

### 三、能力面

工具 18 个（tools.py:2964）+ 参数 Schema 护栏；算法 19 模块（清洗含审计 / Rhinehart 分段 / VIF 共线 / ARX-OE 辨识 / 贝叶斯与 TPE 寻优等）。API：routes.py 聚合 14 个子路由 + main.py 直挂 workspaces 与 fs，共 16 个 routes\_\*.py；门禁：路由层禁直连 DB（gate1=12 存量）、测试裸 drop\_all 须带 \_db\_guard（gate2=1 存量），增量零新增。

### 四、前端

Vue3 工作台：31 条路由（30 个顶层视图 + 6 个向导步骤）；右栏为「寻优子任务 / 寻优监控 / 抉择卡」；图表 ECharts（统计）+ D3（关系与结构，6 组件）。

## 五、质量基线

后端全量 2597 passed + 3 failed (已知环境敏感: CRLF×2 / 文件监听×1) + 2 skipped; 前端 857 例 (844 通过 / 13 存量失败); 双 AST 门禁零新增。

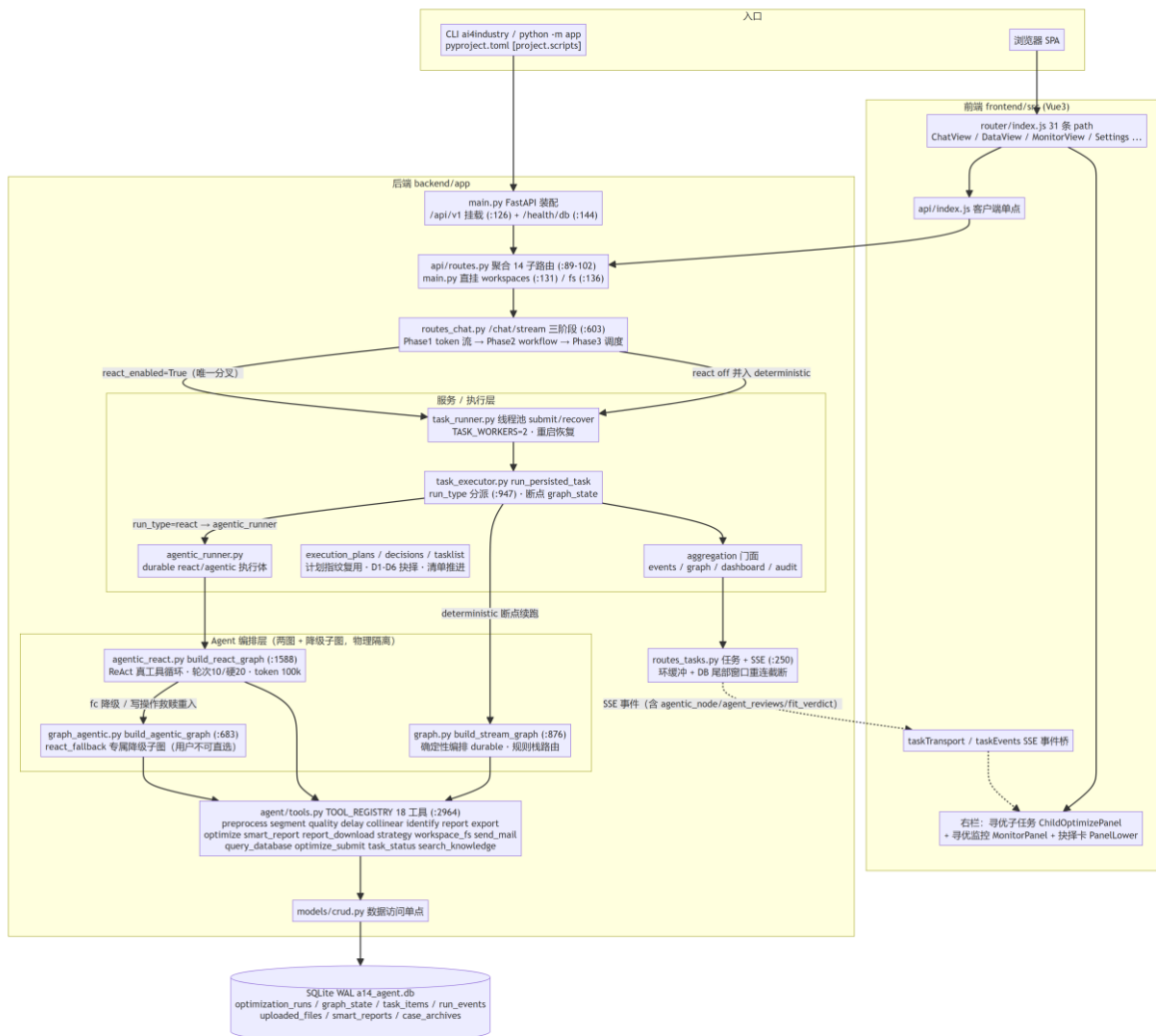


图 1 系统架构全景图 (入口 / API / 编排 / 服务 / 工具 / 数据 + 前端右栏)

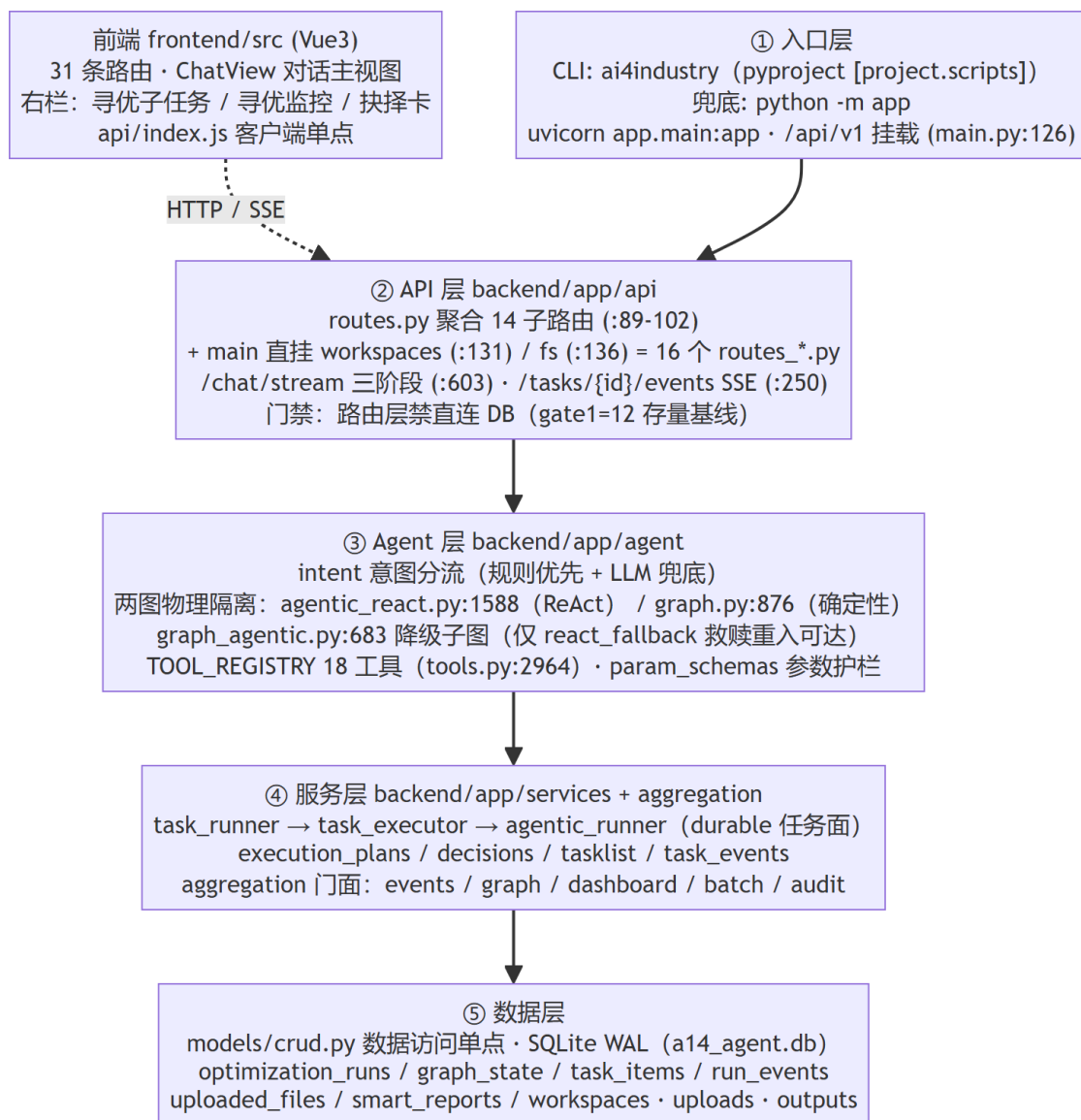


图 2 五层功能地图 (入口 / API / Agent / 服务 / 数据)

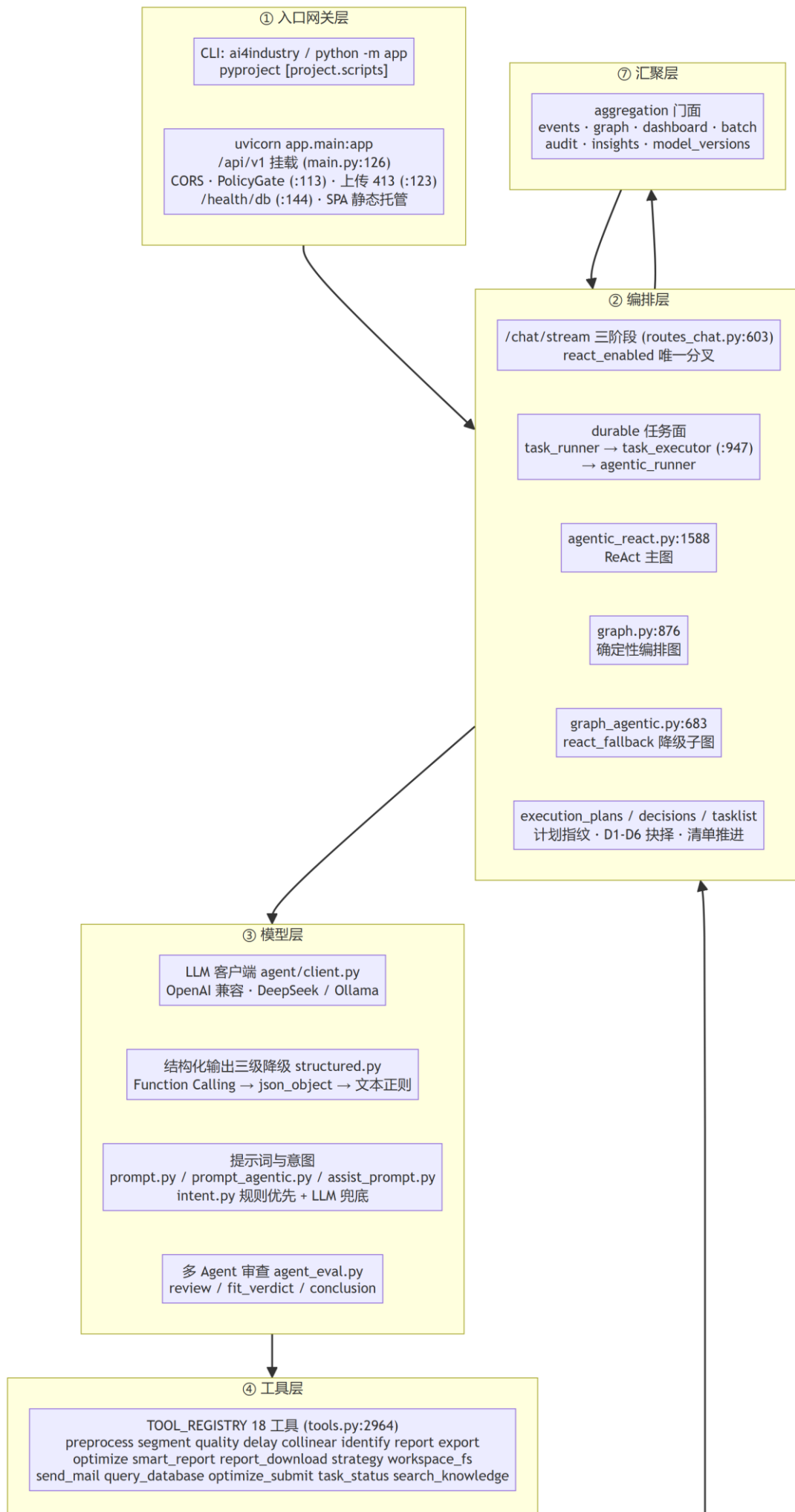


图 3 七层功能地图（入口网关 / 编排 / 模型 / 工具 / 算法 / 记忆 / 汇聚）

多 Agent 审查（代码标准）：按模块映射为 4 类专业 Agent——数据 Agent（preprocess / segment / quality / export / strategy）、分析 Agent（delay / collinear）、建模 Agent（identify / optimize）、报告 Agent（report / smart\_report / report\_download），共覆盖 12 个模块；由 agent\_eval 统一并行审查并给出 FIT 权威结论。