**BDAP大数据通识教育平台**

**概要设计说明书**

|  |  |  |  |
| --- | --- | --- | --- |
| 文档名称 |  | 文档编号 |  |
| 文档类型 | 概要设计说明书 | 版本信息 |  |
| 内部密级 | 公开 | 外部密级 | 自用 |
| 创建人 |  | 创建日期 |  |
| 审批人 |  | 审批日期 |  |

**文档修订历史**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **版本** | **作者** | **版本变化对象** | **变化内容描述** | **审核人** | **批准人** | **批准日期** |
| V1 | 晏成昊 | 2021.10.23 | 创建 | V1 | 晏成昊 | 2021.10.23 |
| V2 | 王孝诚、晏成昊、孟新凯、崔翔冲、张聪、任浩箐、曹晨雨 | 2022.3.23 | 修订 | V2 | 王孝诚、晏成昊、孟新凯、崔翔冲、张聪、任浩箐、曹晨雨 | 2022.3.23 |
| v2.1 | 王孝诚、晏成昊 | 2022.3.24 | 加入数据管理 | v2.1 | 王孝诚、晏成昊 | 2022.3.24 |
| v3.0 | 郭豫宁、张宇轩 | 2024.3.23 | 加入多模态 | v3.0 | 郭豫宁、张宇轩 | 2024.3.23 |
| V4.0 | 于果 | 2025.6.18 | 加入大模型相关功能 | V4.0 | 于果 | 2025.6.18 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**目录**

[1.](#_Toc29456) **[概述](#_Toc29456)** [4](#_Toc29456)

[1.1 开发背景 4](#_Toc17400)

[1.2 需求分析 4](#_Toc2323)

[1.2.1大模型智能生成工作流： 4](#_Toc1723)

[1.2.2数据预处理智能化： 4](#_Toc2442)

[1.2.3模型微调与定制化服务： 5](#_Toc9661)

[1.2.4前端框架重构与状态管理优化： 5](#_Toc23412)

[1.3 系统架构 5](#_Toc29393)

[2.](#_Toc15584) **[模块设计](#_Toc15584)** [7](#_Toc15584)

[2.1. 智能工作流生成 7](#_Toc16683)

[2.1.1 技术路线 8](#_Toc5140)

[2.2. 学生实验行为分析与动态反馈系统 9](#_Toc9841)

[2.2.1 技术路线 9](#_Toc3777)

[2.3. LLM驱动的智能答疑与数据处理引导系统 10](#_Toc30092)

[2.3.1 智能助教技术路线 11](#_Toc10280)

[2.3.2 交互式答疑机制技术路线 11](#_Toc18639)

[2.4.大模型应用开发工具链 12](#_Toc13594)

[2.4.1 技术路线 12](#_Toc24163)

[3](#_Toc19766) **[开发规范设计](#_Toc19766)** [13](#_Toc19766)

[3.1 存储实体设计 13](#_Toc24869)

[3.1.1 用户信息表 13](#_Toc25298)

[3.1.2 准入控制表 14](#_Toc18768)

[3.1.3 实验信息表 14](#_Toc10096)

[3.1.4 组件表 14](#_Toc7090)

[3.1.5 元数据记录表 14](#_Toc2232)

[3.1.6 session表 15](#_Toc17205)

[3.1.7 通信子码设计 15](#_Toc12866)

[3.2 部署环境设置 16](#_Toc24578)

[3.3 软件环境 16](#_Toc8890)

[3.4 接口设计 17](#_Toc151)

[3.5 代码提交规范 17](#_Toc5927)

[4.](#_Toc7281) **[人员安排](#_Toc7281)** [18](#_Toc7281)

[5.](#_Toc25943) **[未来计划](#_Toc25943)** [18](#_Toc25943)

**1. 概述**

**1.1 开发背景**

作为大数据通识教育平台，BDAP3.0在BDAP2.0的基础上迭代，主要增添了图计算、社交关系探索两大功能板块，同时优化了大数据框架，实现了数据一体化存储和并行计算，已经为北邮师生提供了极大的便利。但随着大模型技术的兴起，主要聚焦于传统算数据挖掘算法的BDAP3.0已经不能够满足学生学习适应前沿技术的需要，同时，随着BDAP3.0的推广使用，在高并发、大流量的场景下，服务器亟需减负，数据的复用和压缩传输也都有优化空间，由此，展望BDAP4.0系统。



**1.2 需求分析**

大数据通识教育平台BDAP4.0所面向的对象是大量编程基础较薄弱，仅学习过大数据通识教育的基础理论的初学者，想要利用机器学习、深度学习知识，以图形化方式甚至是自然语言的方式自主构建工作流，并在云端完成数据挖掘流程。

综上，BDAP4.0需要具有以下几个特点：

1. 与大模型技术相结合以跟进新技术前沿
2. 数据高度复用、压缩以满足高并发场景
3. 界面简洁，引导清晰以助力初学者使用

**1.2.1大模型智能生成工作流：**

数据处理与机器学习的复杂性使得工作流设计成为用户的核心痛点。当前，BDAP3.0通过图形化界面降低了技术门槛，但面对大模型时代的多样化任务需求，仍需进一步智能化。大模型智能生成工作流功能旨在通过自然语言交互或预定义模板，自动解析用户意图并生成端到端的工作流。例如，用户输入“训练一个基于用户行为的推荐系统”，系统将自动生成包含数据加载、特征工程、模型训练、评估与部署的完整流程。该功能将结合大模型的意图识别能力，与知识库中的工作流模板进行匹配，并通过多轮交互优化。最终，生成的代码或图形化流程将兼容Spark大数据框架，并支持与BDAP的分布式数据仓库无缝集成，形成标准化的数据挖掘管道。



**1.2.2数据预处理智能化：**

数据预处理是机器学习成功的关键环节，但其难度和复杂性甚至高于机器学习算法本身，常导致用户因技术门槛而放弃优化。BDAP3.0将引入基于大模型的预处理智能化功能，通过自动分析数据分布、缺失值模式及异常值特征，推荐清洗策略。例如，对于冠心病数据，系统可识别重复记录、修正错误时间戳，并基于领域知识构造新特征（如用户身体素质指标）。该功能将结合AutoML技术，动态生成预处理脚本，并通过可视化组件库提供拖拽式操作。

**1.2.3模型微调与定制化服务：**

大模型虽具备通用能力，但针对特定领域任务仍需微调以提升效果。BDAP3.0将提供低代码微调接口，用户可通过上传自定义数据集并配置参数发起训练任务。系统将基于LoRA技术实现高效微调，并通过交互式可视化组件展示训练过程中的损失曲线与准确率变化。微调后的模型可通过Unity Catalog注册并共享，支持docker容器化部署到生产环境。

**1.2.4前端框架重构与状态管理优化：**

随着平台功能的持续扩展，现有前端架构在数据冗余、组件间通信复杂性及状态一致性方面逐渐暴露瓶颈。例如，多个组件需重复获取相同数据，导致冗余请求与内存占用增加；而跨层级的props传递则使得状态更新逻辑难以追踪。为解决此类问题，BDAP3.0将基于Vue3框架与Pinia状态管理库进行前端重构，通过集中化、模块化的状态管理策略，显著减少数据冗余并提升开发效率。

**1.3 系统架构**

BDAP大数据通识教育平台v4.0将集成分布式存储与计算、传统机器学习、图机器学习、深度学习和大模型应用，实现从学生引导、实验答疑到综合教学评价的体系完整的教学系统，平台架构图如图1所示。

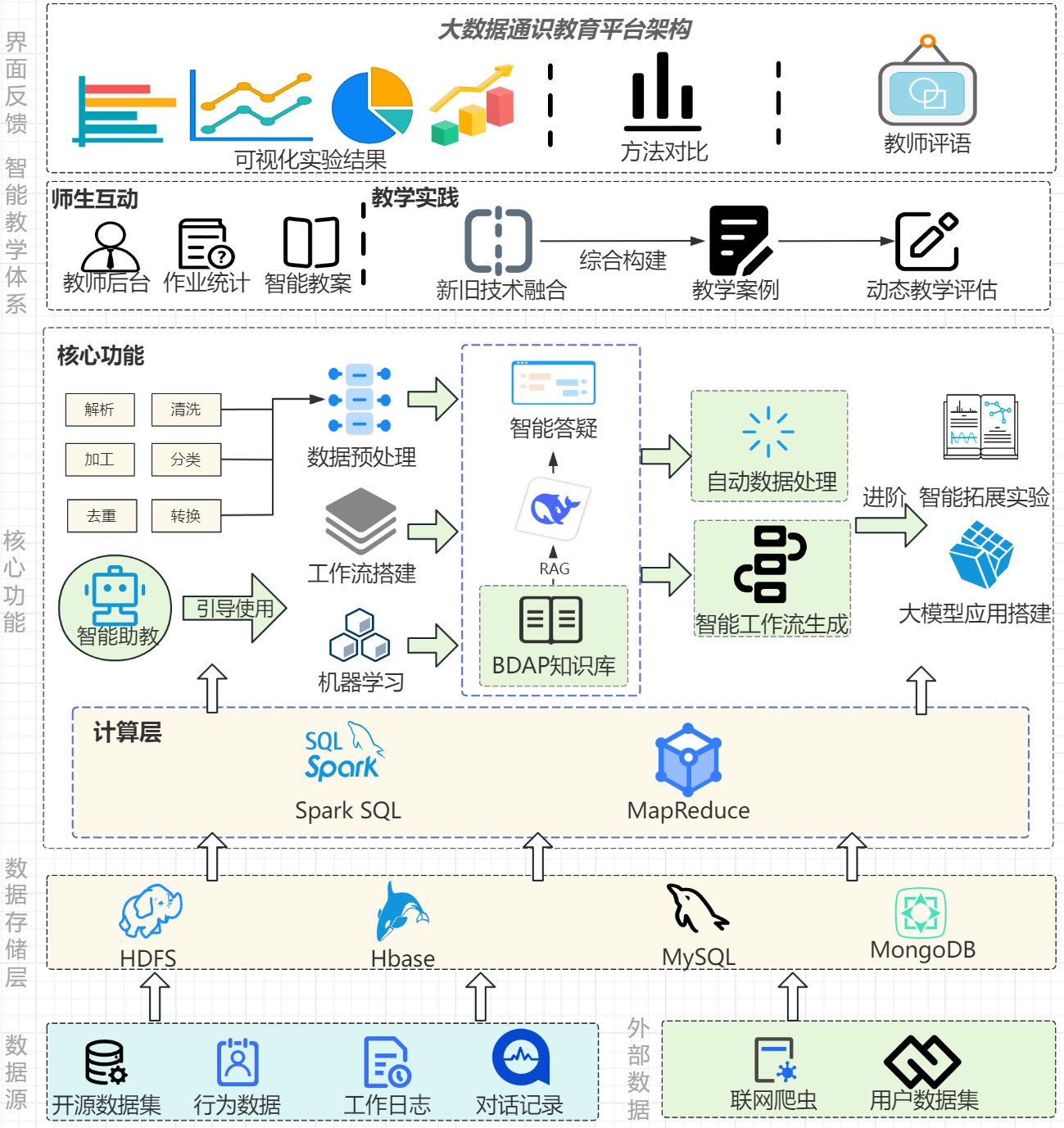


图1（黄色-已完成、绿色-近期完成、蓝色-未来完成）

底层为业务数据源和事件源，包括表格存储、MongoDB、消息队列、视频源和其他文件等内部数据源，以及开源数据集、用户行为数据、工作流日志、模型对话记录、等外部数据源。此外，还有联网爬虫和用户数据集作为补充。



数据存储部分，提供了HDFS、Hbase为基础的分布式文件系统，支持在线读取和内存计算，同时，也为平台后端通过与MySQL、MongoDB的交互，能够同时处理大量并发请求，维护用户实验数据。



核心技术部分，计算层包含离线计算工具如SQL Spark和Map/Reduce等，并支持智能化传统数据挖掘流程，提供AI赋能的数据挖掘功能如自动数据处理、智能工作流生成和大模型应用搭建。

在以上技术的基础上，BDAP4.0还搭载了作业统计、教师后台、智能教案功能，提供了融合新旧技术，体现技术迭代的易上手的教学案例，且能够智能分析学生特点，智能生成拓展实验，助学生加深对大数据技术的理解。同时，平台也提供了简洁直观的UI界面，支持多种数据结果进行可视化，方便学生多角度对比不同方法的性能。在教学的最后，平台可以根据学生的综合表现(包括但不限于技术基础、求知精神、方法创新、实验结果等)，进行全面地教学评价。

1. **模块设计**

**2.1 继承功能**

基于对BDAP 2.0平台的深入分析，我为您整理出核心功能和技术架构：

**2.1.1. 可用组件构成**

（1）机器学习组件库



提供完整的机器学习算法组件，支持从数据预处理到模型训练的端到端流程。

技术实现：

传统机器学习：基于Spark MLlib实现线性回归、决策树、SVM、KNN、聚类等算法

深度学习：基于PyTorch实现CNN、RNN、LSTM等神经网络模型

推荐系统：协同过滤、关联规则挖掘算法

特征工程：TF-IDF、特征选择、数据标准化组件

（2）图计算组件库



支持复杂图算法和网络分析，提供图数据的处理、分析和可视化能力。

技术实现：

图算法：基于Spark GraphX实现社区发现、最短路径、中心性分析

图生成：随机图、小世界网络、无标度网络生成算法

图可视化：基于D3.js的交互式网络图展示

（3）数据处理组件



提供数据清洗、转换、聚合等基础数据处理能力。

技术实现：

数据清洗：缺失值处理、异常值检测、重复值删除

数据转换：类型转换、编码转换、格式转换

数据聚合：分组统计、窗口函数、数据透视

**2.1.2 文件管理系统**

（1）分布式文件存储



功能描述：基于HDFS的分布式文件存储系统，支持大文件的存储和管理。

技术实现：

HDFS集成：深度集成Apache HDFS，支持PB级数据存储

多租户隔离：用户级文件命名空间隔离，防止越权访问

权限控制：基于ACL的细粒度权限管理

（2）文件操作引擎

功能描述：提供文件上传、下载、预览、管理等完整功能。

技术实现：

流式传输：支持大文件的分块上传和断点续传

格式支持：自动识别CSV、JSON、Parquet、图片、视频等格式

预览功能：文本分页预览、图片缩略图、视频关键帧提取

**2.1.3. 平台管理系统**

（1）用户权限管理

功能描述：完整的用户生命周期管理和权限控制体系。

技术实现：

JWT认证：基于JWT Token的无状态身份认证

RBAC权限：基于角色的访问控制模型

多级权限：用户级、角色级、操作级权限控制

3.2 资源监控管理

功能描述：系统资源监控、任务调度和性能优化。

技术实现：

服务监控：基于Spring Boot Actuator的系统监控

负载均衡：基于Consul的服务发现和负载均衡

任务调度：智能的任务调度和资源分配算法

**2.1.4. 前后台交互机制**

（1）API网关设计

功能描述：统一的API入口，负责路由转发、安全认证、流量控制。

技术实现：

Spring Cloud Gateway：基于Spring Cloud Gateway的API网关

智能路由：基于路径和规则的服务路由转发

安全防护：JWT验证、权限检查、请求过滤

（2）服务间通信

功能描述：微服务间的同步和异步通信机制。

技术实现：

Feign客户端：声明式的HTTP服务调用

消息队列：基于RabbitMQ的异步消息传递

熔断器：Hystrix熔断器实现服务降级

**2.1.5. 可视化系统**

（1）数据可视化



功能描述：丰富的数据可视化展示能力，支持多种图表类型。

技术实现：

ECharts集成：基于ECharts的图表渲染引擎

交互式操作：缩放、筛选、钻取等交互功能

实时更新：基于WebSocket的实时数据更新

（2）实验流程可视化



功能描述：实验工作流的图形化展示和实时监控。

技术实现：

D3.js渲染：基于D3.js的工作流图形渲染

状态同步：实时显示节点执行状态和进度

结果展示：实验结果的图表化展示和分析

**2.1.6. 执行引擎架构**

（1）多引擎支持

功能描述：支持多种计算引擎，根据任务特点自动选择最优引擎。

技术实现：

Spark引擎：基于Apache Spark的大数据处理



Ray引擎：基于Ray的分布式计算框架



PyTorch引擎：基于PyTorch的深度学习训练



（2）任务调度系统



功能描述：智能的任务调度和资源管理。

技术实现：



Livy会话管理：基于Apache Livy的Spark会话管理

优先级调度：基于任务优先级的智能调度算法

容错机制：任务失败自动重试和故障恢复

**2.2 迭代升级**

**2.2.1. 智能工作流生成**

（1）BDAP平台计划开发一个动态工作流生成模块，同时在文本处理模块、视频关系抽取、图机器学习模块中引入大模型辅助数据预处理组件，提供提示词模板。同时，提供纯大模型方法，以便学生直观感受到新旧技术各自的优缺点。

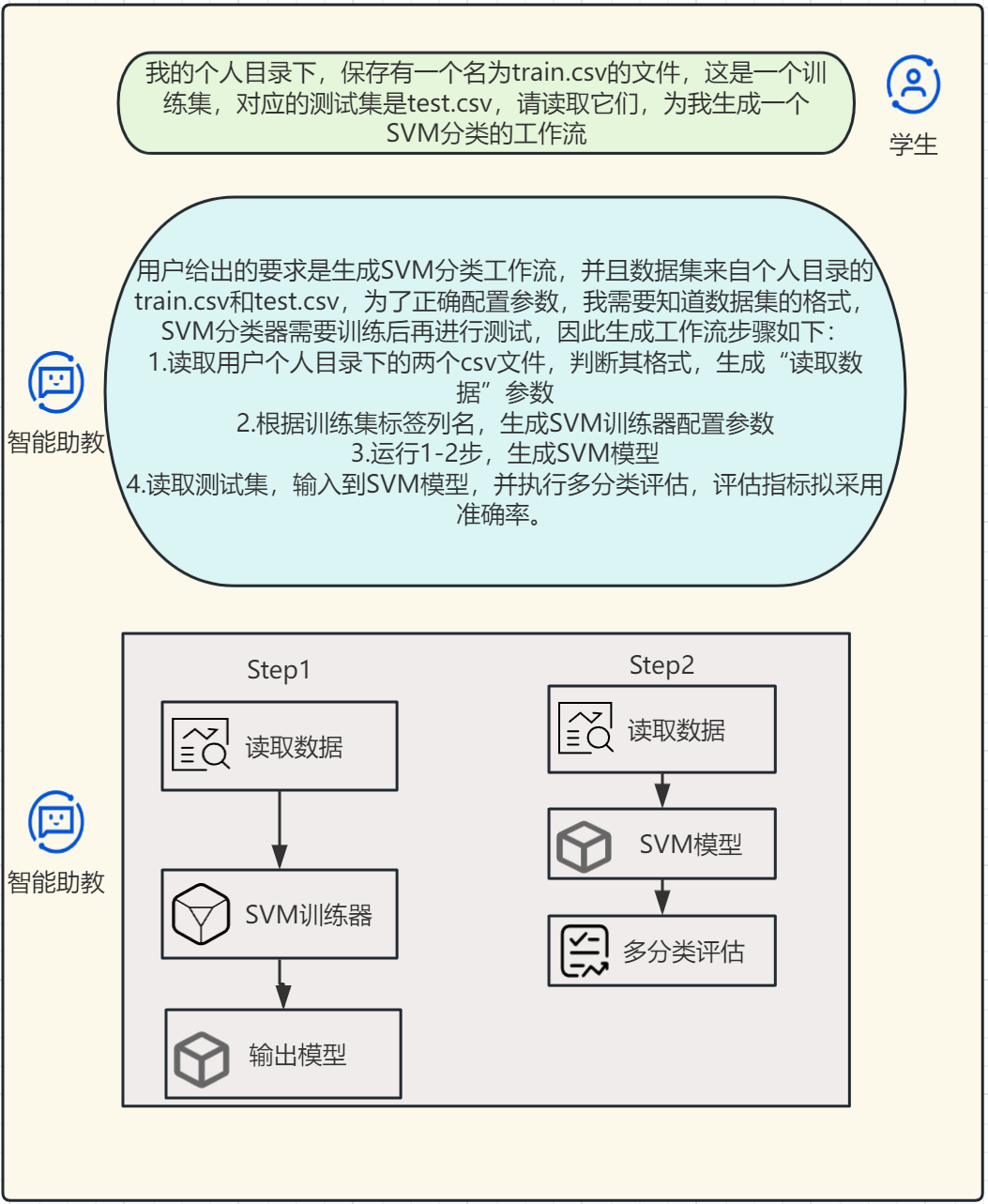


图2 智能工作流生成与思维链



1. 自然语言解析与意图识别



核心技术：基于预训练语言模型的自然语言理解能力，结合领域适配的意图识别模型。

实现细节：

（1）意图分类：通过微调预训练模型，识别用户输入中的任务类型。

（2）实体提取：提取关键参数（如数据源类型、目标变量、特征维度）。

（3）上下文建模：利用Transformer架构对长文本进行语义建模，捕捉用户需求中的隐含逻辑。

2. 工作流结构生成

核心技术：基于结构化知识库的模板匹配与规则引擎，结合生成式模型的动态组合能力。

实现细节：

（1）模板库构建：构建领域特定的结构化知识库，包含典型任务的标准化工作流模板。

（2）每个模板关联任务类型、适用场景、关键步骤及参数。

（3）动态组合逻辑：根据用户需求，以余弦相似度计算匹配最相似的模板，并结合规则引擎Drools调整步骤顺序或替换算法。利用LLM扩展模板边界，生成非典型但合理的流程。

3.思维链生成

核心技术：基于提示词工程与序列生成模型（例如T5、BART）的Step-by-Step解释生成。



实现细节：

（1）提示词模板设计：定义标准化的CoT生成模板，示例：

任务：{任务类型}

步骤：{步骤名称} → 目的：{解释目的}，方法：{方法选择依据}

通过微调模型学习模板中的逻辑关联（如“数据清洗”的目的是“去除噪声以提高模型精度”）。

（2）动态解释生成：对于每一步骤，生成自然语言解释。结合知识库中的最佳实践案例，提供决策依据。

4. 预训练模型微调与知识库适配

核心技术：基于LoRA或全量微调的领域自适应技术，结合结构化知识库的监督学习。



实现细节：

（1）数据准备：构建微调数据集，包含任务描述、对应工作流模板及CoT解释(数据标注需覆盖多任务类型)。

（2）模型训练策略：微调目标是优化模型对领域术语的理解、任务逻辑的推理能力。损失函数设计采用多任务学习框架，联合优化意图识别、步骤生成与CoT解释的损失。



（3）知识库整合：将结构化知识库转化为模型可训练的向量表示，作为微调时的辅助输入。在推理阶段，通过检索增强生成技术，结合知识库中的案例动态调整生成结果。



5. 系统集成与评估



技术实现：

（1）API接口：将自然语言解析、工作流生成与CoT解释封装为RESTful API，供BDAP平台调用。

（2）实时反馈机制：允许用户对生成的工作流提出修改建议，触发模型的在线学习更新。

评估指标：

（1）准确性：工作流步骤与用户需求的匹配度。

（2）CoT质量：解释文本的清晰度与合理性。

（3）泛化能力：模型对未见任务类型的适应性。

**2.2.2. 学生实验行为分析与动态反馈系统**

(1)开发实验数据采集模块，实时记录学生实验操作日志（如代码提交、错误类型、实验进度等）。

(2)构建动态学情分析引擎，通过自然语言处理（NLP）和机器学习模型分析学生行为数据，识别学习难点（示例：“训练集划分不理解”“参数配置错误频繁”）。



(3)设计可视化教师后台，以热力图、趋势图等形式展示学生群体及个体的共性问题，辅助教师制定针对性教学方案。



1. 实验数据采集模块

（1）核心技术：实时日志记录与事件监控。

（2）实现细节：多源数据整合：通过API接口或SDK嵌入实验平台，实时捕获代码提交、运行错误、调试操作等行为日志。



（3）结构化存储：将非结构化日志通过NLP解析为标准化字段（如“错误类型”“操作时长”），并实时写入数据库。



（4）实时传输：利用WebSocket或Kafka实现低延迟数据传输，确保教师后台能即时响应学生操作。



2. 动态学情分析引擎

核心技术：NLP与机器学习模型联合建模。

实现细节：

（1）难点识别：基于预训练语言模型提取学生错误日志的语义特征，结合分类模型识别高频问题。

（2）动态反馈生成：针对识别出的难点，调用知识库中的教学模板，生成自然语言解释或代码片段建议。

（3）实时性优化：采用流式计算框架Flink处理实时行为数据，确保分析结果在秒级内反馈。

3. 可视化教师后台

核心技术：交互式可视化与多维数据聚合。



实现细节：

（1）群体热力图：通过ECharts或D3.js绘制学生错误类型分布热力图，高亮共性问题。



（2）个体趋势图：为每位学生生成学习曲线，结合聚类分析识别进步/退步群体。



（3）实时更新：通过WebSocket推送最新分析结果，教师可即时调整教学策略。



**2.2.3. LLM驱动的智能答疑与数据处理引导系统**



(1)集成大语言模型，开发“智能助教”，支持学生上传数据集并提出问题。

(2)设计交互式对话流程，LLM通过上下文理解生成分步骤指导，并提供代码片段示例。

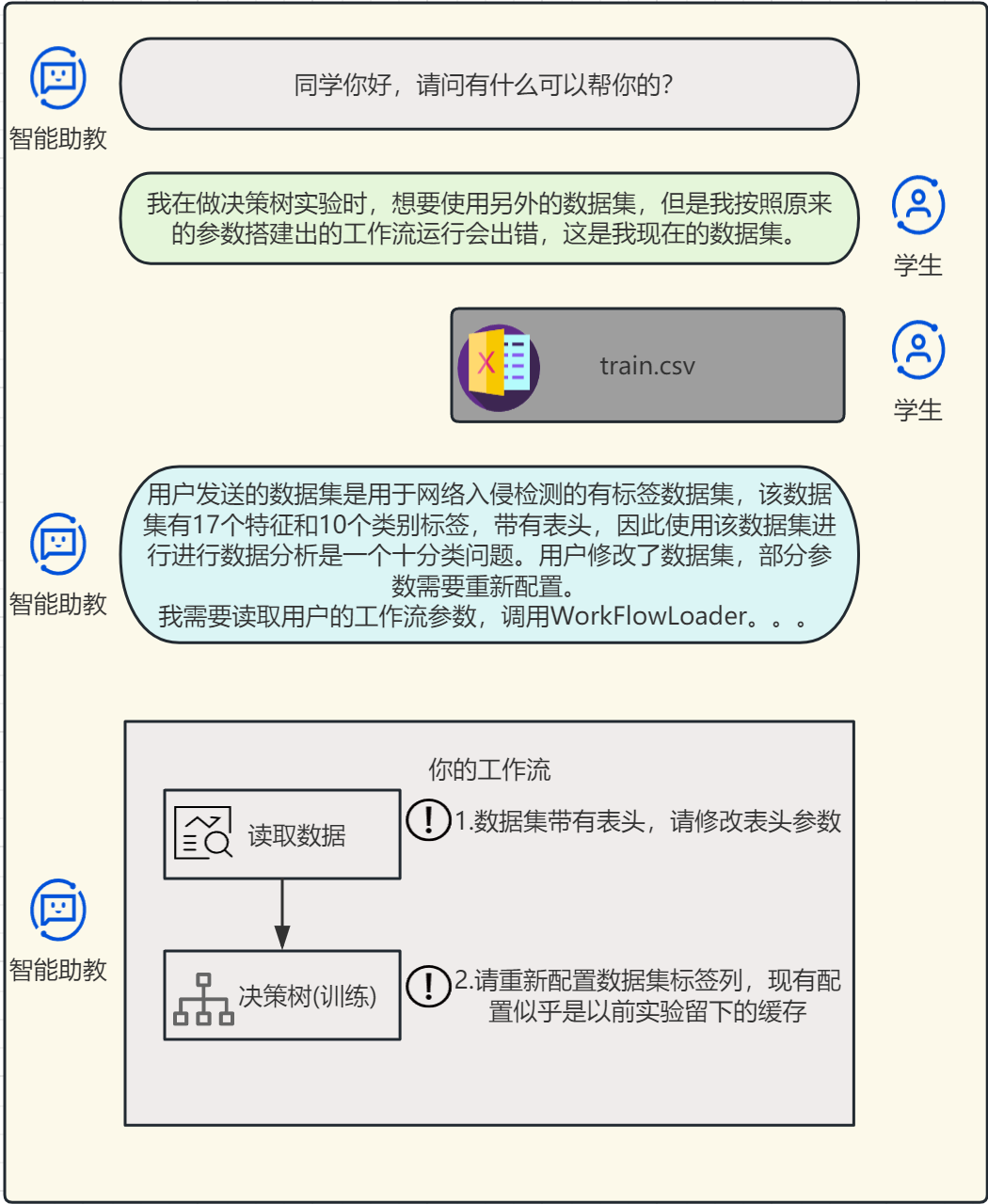


图3 LLM驱动的智能答疑

核心技术：预训练语言模型的定制化部署与多模态输入支持。



实现细节：

（1）模型选择与优化：选用开源模型或企业级模型作为基础，通过LoRA微调适配教育场景。

（2）数据上传与上下文建模：支持学生上传结构化数据或非结构化数据，通过NLP解析生成元数据描述。

（3）代码生成与验证：采用“自然语言→代码模板→参数填充”的三层生成逻辑。例如，用户提问“如何标准化数值特征？”，模型首先选择标准化模板，再根据数据类型填充参数。



核心技术：对话状态管理与多轮交互优化。



实现细节：

（1）上下文感知的分步骤指导：基于LangChain的ChatPromptTemplate，构建包含全局指令与动态消息的提示模板。例如，当用户提问“如何可视化缺失值？”时，模型生成分步指令：

步骤1：加载数据并统计缺失值比例

步骤2：选择可视化方式

步骤3：执行代码并调整参数

利用检索式NLG（Retrieval-based Generation，NLG），从预定义的知识库中召回最佳实践案例，增强指导的领域相关性。

**2.2.4.大模型应用开发工具链**

大模型应用开发工具链包含以下内容：

(1)大模型适配器：提供预训练模型微调框架，支持学生在特定任务上快速微调模型。



(2)知识库构建组件：集成数据标注工具与向量数据库（一般是FAISS），学生可上传领域知识并构建个性化知识库。



(3)工具组件库：封装常用工具，支持学生自由组合构建端到端应用。

1.大模型适配器（Adapter方式）



核心技术：基于LoRA的参数高效微调框架。



实现细节：

（1）模型适配：在预训练大模型的注意力层中插入低秩适配器模块，通过冻结原始参数，仅训练低秩矩阵。利用transformers库的PeftConfig和get\_peft\_model接口快速加载适配器配置，支持动态切换不同任务的适配器权重。

（2）任务微调：针对文本分类任务，构建带标签的微调数据集，通过Trainer接口进行参数高效训练。

（3）优化策略：采用混合精度训练和梯度累积，减少显存占用(目标：小于8GB)。

（4）部署与导出：微调完成后，将适配器权重与原始模型合并，生成轻量级部署包，支持通过TorchScript或ONNX格式部署到服务器。

2.知识库构建组件



核心技术：数据标注自动化与向量数据库(FAISS)集成。



实现细节：

（1）数据标注与预处理：使用label-studio工具提供图形化标注界面，支持PDF/Word文档的实体标注。

（2）自动化标注：基于Easy-dataset框架，结合预训练语言模型生成初始标注结果，人工校验后形成结构化数据。



（3）向量数据库构建：通过FAISS库构建近似最近邻索引，支持大规模向量存储与快速检索。嵌入模型使用sentence-transformers的all-MiniLM-L6-v2模型，将文本转换为768维向量并存储至FAISS向量库。



（4）检索增强生成（RAG）：结合LangChain的VectorStoreRetriever，实现从知识库中检索相关文档并注入模型输入。例如，用户提问“如何处理缺失值？”时，系统检索知识库中“缺失值填充方法”的向量文档并生成解释。



3.工具组件库

核心技术：模块化封装与端到端流程支持。

实现细节：

（1）数据可视化组件：封装ECharts和Plotly的核心API，提供开箱即用的图表组件。例如，一个组件生成缺失值分布热力图。



（2）支持交互式探索：用户可点击图表元素触发数据钻如查看该时间段的详细日志。



（3）模型评估组件：集成scikit-learn的评估指标，封装为ModelEvaluator类，自动计算多分类指标并生成混淆矩阵。



（4）提供模型比较功能：支持并排对比多个模型的ROC曲线、PR曲线，辅助学生选择最优模型。



（5）端到端流程支持：通过Pipeline设计模式，允许学生将数据预处理、模型训练、评估组件按需组合。提供Jupyter Notebook模板，引导学生从数据加载到部署的全流程实践。



**3 开发规范设计**

**3.1 存储实体设计**

**3.1.1 用户信息表**

|列名 | 列描述| 数据类型|是否可以为空|备注|

|:-:| :-:|:-:|:-:|:-:|

|id|主键| String | N| |

|userId |用户id |String |N |字母数字下划线组成 |

|userName |用户名称 |String |Y | |

|password |密码 |String |N | |

|role| 用户角色| String|N |普通用户/admin/root |

**3.1.2 准入控制表**

|列名 | 列描述| 数据类型|是否可以为空|备注|

|:-:| :-:|:-:|:-:|:-:|

|id|主键| String | N| |

|userId |用户id |String |N |字母数字下划线组成 |

|refreshToken |令牌 |String |N | |

|expireAt|令牌过期时间 |String |N |YYYY-mm-dd hh:mm:ss |

**3.1.3 实验信息表**

1. 实验信息表（记录实验信息、工作流结构、类型等）

|列名 | 列描述| 数据类型|是否可以为空|备注|

|:-:| :-:|:-:|:-:|:-:|

|id|主键| String | N| |

|userId |用户id |String |N |字母数字下划线组成 |

|title |实验名称 |String |N | |

|desc |实验内容描述 |String |N | |

|serviceType |实验类型 |String |N | ml/graph 区分结构化数据挖掘和图数据挖掘实验|

|visibility |可见性 |String |N |private/public |

|nodes |工作流结构 |List\<JsonObject> |N | 每个JsonObject中记录了节点组件信息以及设置的参数|

|createdTime |实验创建时间 |long |N |时间戳 |

|updatedTime |实验修改时间 |long |N |时间戳 |

**3.1.4 组件表**

|列名 | 列描述| 数据类型|是否可以为空|备注|

|:-:| :-:|:-:|:-:|:-:|

|id|主键| String | N| |

|name |组件名 |String |N | |

|group |组件所属类别 |String |N |用于前端创建组件目录树 |

|numOfInput |组件输入节点数 |Integer |N | |

|numOfOutput |组件输出节点数 |Integer |N | |

|size |组件展示大小 |String |Y |默认 200x40 |

|simpleAttributes |组件需要设置的简单属性列表 |List\<JsonObject> |N | 每个JsonObject中记录了参数名称、参数类型、可选参数等信息|

|complicatedAttributes |组件需要设置的复杂属性列表 |List\<JsonObject> |N |每个JsonObject中记录了和该组件相关的自定义参数对象 |

|serviceType |组件类型 |String |N | ml/graph 区分组件的可用范围|

|createdTime |组件创建时间 |long |N |时间戳 |

|updatedTime |组件修改时间 |long |N |时间戳 |

**3.1.5 元数据记录表**

|列名 | 列描述| 数据类型|是否可以为空|备注|

|:-:| :-:|:-:|:-:|:-:|

|id|主键| String | N| |

|expInfoId |实验Id |String |N | |

|moduleId |组件id |String |N | |

|moduleMark |组件标示 |String |N |通过expInfoId+moduleId+moduleMark唯一标示某个组件 |

|userId |用户id |String |N | |

|metaData |元数据信息 |JsonObject |N |经过该组件处理，输出的数据的表头信息 |

|createdTime |创建时间 |long |N |时间戳 |

|列名 | 列描述| 数据类型|是否可以为空|备注|

|:-:| :-:|:-:|:-:|:-:|

|id|主键| String | N| |

|filePath |文件路径 |String |N | |

|metaData |元数据信息 |JsonObject |N |文件的表头信息 |

|createdTime |创建时间 |long |N |时间戳 |

|fileModTime |修改时间 |long |N |时间戳 |

**3.1.6 session表**

|列名 | 列描述| 数据类型|是否可以为空|备注|

|:-:| :-:|:-:|:-:|:-:|

|id|主键| String | N| |

|userId |用户id |String |N | |

|expId |实验id |String |N | |

|kind |session类型 |String |N |normal/deeplearning 区分在cpu机器和gpu机器上创建session |

|livyAddr |session分配的地址 |String |N | |

|sessionId |会话id |String |N | |

|isDeleted |是否有效 |String |N | 软删除字段|

|createdTime |创建时间 |long |N |时间戳 |

|fileModTime |修改时间 |long |N |时间戳 |

**3.1.7 通信子码设计**

| 通信子码 | 异常信息 |

| :-: | :-: |

| 4000| 广义上的错误请求

| 4001 | 请求参数不合法

| 4002| 请求格式不合法

| 4003| 无效的URL

| 4004| 非法token

| 4005| 路径异常

| 4006| 文件已经存在

| 4010| 鉴权失败

| 4011| access\_token过期

| 4012| refresh\_token过期

| 4031| 密码不正确, 禁止访问

| 4032| 在指定时间范围之外提交作业, 禁止访问

| 4041| 用户不存在

| 4061| 禁止更改root的权限

| 4080| 请求超时

| 4151| 文件不是csv后缀

| 5000| 数据库记录查找失败

| 5001| 异常

| 5002| jwt解码异常

| 5003| parse异常

| 5004| IO异常

| 5005| hdfs路径或IO异常

| 5006| Json处理异常

| 5007| 批量删除部分失败

| 5030| 系统繁忙, 请稍后再试

**3.2 部署环境设置**

| IP地址                        | 内容                |

| ----------------------------- | ------------------- |

| bdap-cluster-01:9870          | Hadoop Overview     |

| bdap-cluster-01:8088          | Spark工作流         |

| bdap-cluster-01:7180          | Cloudrea Manager    |

| bdap-cluster-02:8501          | Spring Boot Admin   |

| bdap-cluster-02:8090          | Confluence          |

| bdap-cluster-02:8091          | Jira                |

| bdap-cluster-02:3000          | BDAP Admin BDAP后台 |

| bdap-cluster-03-04/08-10:8998 | livy sessions       |

| bdap-cluster-08:8502          | docker harbor   用户名admin 密码bdap123456  |

| bdap-cluster-08:30001         | k8s Dashboard       |

| bdap-cluster-08:31000         | k8s consul          |

| bdap-cluster-08:30601         | 后端gateway端口      |

**3.3 软件环境**

| 类型           | 名称                                                 | 版本      | 语言 | 用途                                                         |

| :------------- | :--------------------------------------------------- | :-------- | :--- | :----------------------------------------------------------- |

| 操作系统       | Centos                                               | 7.8       | 英文 | 服务器搭载的操作系统                                         |

| GPU驱动        | Nvidia Driver                                        | 460.67    | 英文 | GPU所需的驱动程序                                            |

| Hadoop集群环境 | CDH(Cloudera's Distribution Including Apache Hadoop) | 6.2.1     | 英文 | Hadoop大数据集群基础环境，包含HDFS,YARN,Spark,HBase,ZooKeeper等大数据组件 |

| Java环境       | JDK(Java Development Kit)                            | 1.8.0\_144 | 英文 | Java运行和开发环境                                           |

| Java环境       | Maven                                                | 3.6.3     | 英文 | Java项目构建系统                                             |

| Java环境       | Nexus                                                | 3.28.1    | 英文 | Maven仓库管理器                                              |

| Python环境     | Miniconda3                                           | 4.9.2     | 英文 | Python运行环境                                               |

| Spark运行环境  | Livy                                                 | 0.7.0     | 英文 | 用于通过REST接口提交Spark作业                                |

| 数据库         | MongoDB                                              | 4.0.10    | 英文 | 用于存储平台需要和产出的相关数据                             |

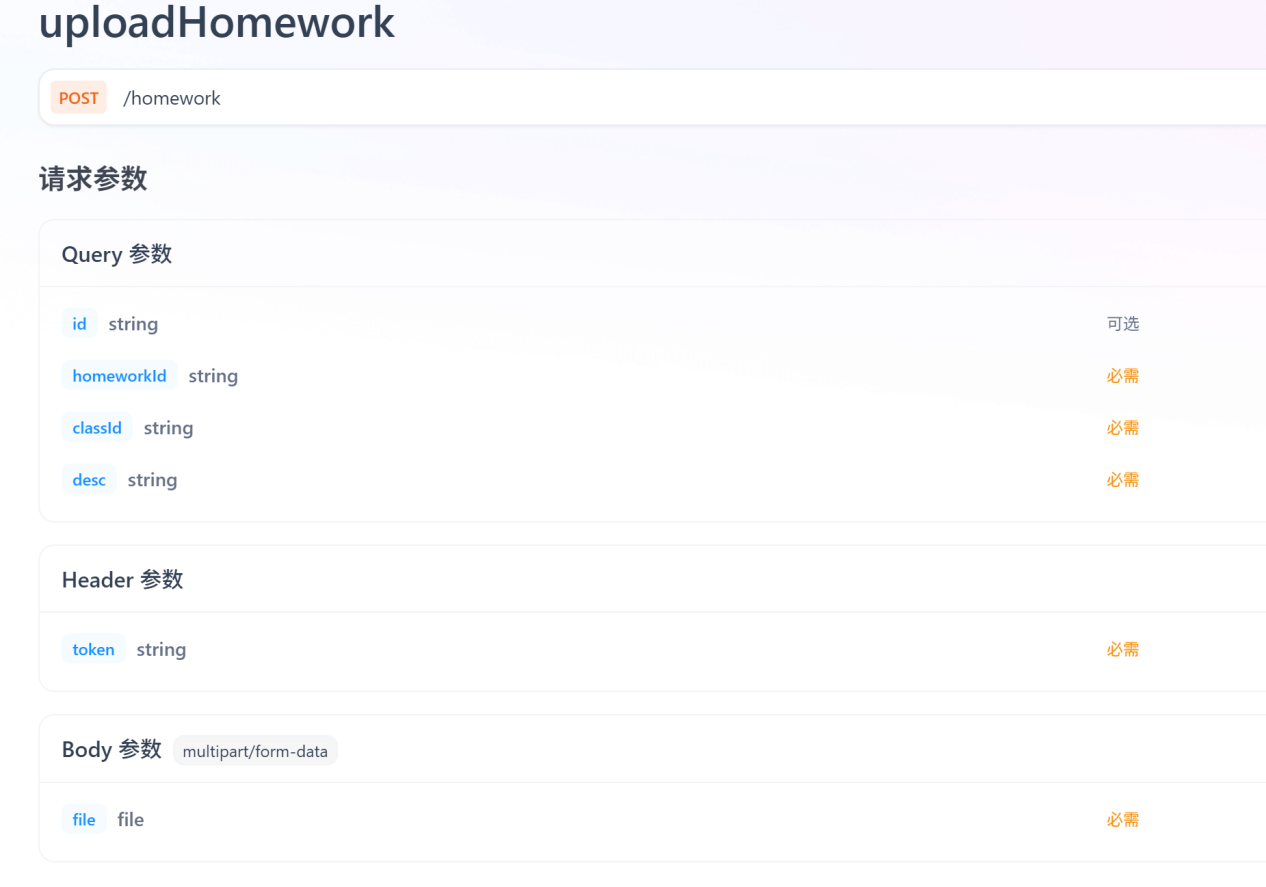
| 前端运行环境   | Node.js                                              | 14.9.0    | 英文 | 用于为平台的前端提供运行环境                                 |

| 在线编程环境   | JupyterHub                                           | 1.3.0     | 英文 | 为平台提供在线编程环境                                       |

| 版本控制       | Git                                                  | 1.8.3.1   | 英文 | 用于控制平台的代码版本                                       |

**3.4 接口设计**

BDAP4.0的开发遵循前后端分离开发的原则，开发人员在开发时应留下符合规范的接口文档，以便前后端对接和后来人接手，接口文档示例图如下。





BDAP的接口已经统一整理发布到<https://izswhhzg9n.apifox.cn/> ，密码123456，如有需要可自行访问，这里由于篇幅问题不再占用版面。

**3.5 代码提交规范**

格式 [service] type:description

```

1. service

   <br> 这个字段标识我们的提交是针对某个服务的，也是为了便于之后在 git 记录中进行筛选。

   考虑到我们的服务名太长，规定五个服务的简称如下

   ```

   login-service: [U]

   livy-service: [L]

   gateway: [G]

   filesystem-service: [F]

   experiment-service: [E]

LLM-service: [L]

   之后如果新起服务再在此进行补充

   ```

2. type

   <br> 标识提交的类别，我们规定下列几种标识

   ```

   1. ADD: 标识添加了新功能

   2. FIX: 修复bug

   3. MOD: 对某些已有功能的更新/迭代

   4. STYLE：代码重构

   5. DEL：删除代码

   ```

3. description

   <br> 对提交内容的简短描述

**4. 人员安排**

1 Vue3新前端

* 数据复用、Pinia统一状态管理：于果、杨组俨
* 新组件UI：于果、徐笑天、周名赫
* LLM聊天模块：王俊超（大四毕设）

2 大模型微调与分布式交互：于果、张旭睿

3 BDAP知识库增强大模型服务部署：齐子昂、杨组俨

4 学生行为监测日志与数据库交互：齐子昂、徐笑天

5 工作流模板设计：于果、张欣怡、徐笑天

6 大模型格式规范与提示模板：齐子昂、徐笑天、张旭睿

7 BDAP维护以及平台迭进：于果、齐子昂、徐笑天、杨组俨

8 运维测试：于果、齐子昂、徐笑天

**5. 未来计划**

1. Vue3 新前端

具体计划：

1. 文件系统和用户登录功能重建(已完成)

2）25.6复现原前端全部功能

1. 使用BDAP知识库增强大模型在大数据通识教育领域的表现

具体计划：

1）25.6调整好大模型输出规范，部署到平台

1. 工作流模板设计与智能生成工作流

具体计划：

1）25.6-25.7完成工作流模板设计，并设计好提示词，同时使用Adapter完成大模型微调

2）25.8完成使大模型按照模板输出回答

3）25.9完成大模型智能生成工作流

1. 智能数据预处理

具体计划：

1）25.7完成专用于理解数据集的大模型微调，并设计数据集处理方案

2）25.8完成大模型智能处理数据

3）25.9设计对应前端界面

1. 智能助教与学生实验监测

具体计划：

1）25.7完成学生实验日志监测，并完成数据库通信

2）25.8-25.9完成学情分析和教师后台

1. 联调线上测试

具体计划：

1）25.8 完成线上联调

2）25.9 完成压力测试