

密 级	内部
阶段标记	J
版 本	1.0.0

# 材料实验数据的自动分析平台功能支撑模块 用户使用手册

(HIAT-2024302B004-SC-01-1.0.0)

湖南先进技术研究院

二〇二五年五月

# 材料实验数据的自动分析平台功能支撑模块

## 用户使用手册

编制 \_\_\_\_\_ 年 月 日

校对 \_\_\_\_\_ 年 月 日

审核 \_\_\_\_\_ 年 月 日

批准 \_\_\_\_\_ 年 月 日

会 签								
	标准							
顾客								
单位	签字	日期						
文件页数	共 93 页							
				标记	处数	更改单号	签字	日期



## 目 录

1 引 言 .....	1
1.1 文档范围和读者 .....	1
1.2 约定和术语 .....	1
2 材料实验数据的自动分析平台功能支撑模块 .....	2
2.1 平台简介 .....	2
2.2 登录 .....	2
2.3 数据智能标注 .....	3
2.3.1 查看数据智能标注模块 .....	3
2.3.2 创建标注项目 .....	4
2.3.3 标注数据 .....	6
2.3.4 标注结果导出 .....	7
2.3.5 标注任务查询 .....	8
2.3.6 删除标注项目 .....	9
2.4 流水线模板 .....	10
2.4.1 查看流水线模板模块 .....	10
2.4.2 新建流水线 .....	11
2.4.3 配置流水线 .....	11
2.4.4 组件介绍 .....	14
2.4.5 编辑流水线 .....	18
2.4.6 复制流水线 .....	18
2.4.7 删除流水线 .....	19

2.5 实验	20
2.5.1 查看实验模块	20
2.5.2 新建实验	21
2.5.3 编辑实验	22
2.5.4 运行实验	22
2.5.5 实验详情查看	23
2.5.6 实验过程参数变化查看	25
2.5.7 终止实验	27
2.5.8 删除实验	27
2.6 自动机器学习	28
2.6.1 查看自动机器学习模块	28
2.6.2 新建实验	29
2.6.3 编辑实验	30
2.6.4 复制实验	31
2.6.5 运行实验	31
2.6.6 实验详情查看	32
2.6.7 终止实验	34
2.6.8 删除实验	35
2.7 超参数自动寻优	36
2.7.1 查看超参数自动寻优模块	37
2.7.2 新建实验	38
2.7.3 编辑实验	38

2.7.4	复制实验	39
2.7.5	运行实验	40
2.7.6	实验详情查看	40
2.7.7	终止实验	42
2.7.8	删除实验	43
2.8	主动学习	44
2.8.1	查看主动学习模块	44
2.8.2	新建实验	45
2.8.3	编辑实验	46
2.8.4	复制实验	46
2.8.5	运行实验	47
2.8.6	实验详情查看	48
2.8.7	终止实验	50
2.8.8	删除实验	51
2.9	数据集管理	52
2.9.1	查看数据集管理模块	52
2.9.2	新建数据集	53
2.9.3	查看数据集	54
2.9.4	创建新版本	55
2.9.5	版本对比	55
2.9.6	下载数据集	56
2.9.7	删除版本	57

2.9.8 删除数据集 .....	58
2.10 模型管理 .....	59
2.10.1 查看模型管理模块 .....	59
2.10.2 新建模型 .....	60
2.10.3 查看模型 .....	61
2.10.4 创建新版本 .....	63
2.10.5 版本对比 .....	63
2.10.6 下载模型 .....	64
2.10.7 删除版本 .....	65
2.10.8 删除模型 .....	66
2.11 镜像管理 .....	67
2.11.1 查看镜像管理模块 .....	67
2.11.2 创建镜像 .....	68
2.11.3 查看详情 .....	69
2.11.4 新增镜像版本 .....	69
2.11.5 删除版本 .....	70
2.11.6 删除镜像 .....	71
2.12 代码管理 .....	72
2.12.1 查看代码管理模块 .....	72
2.12.2 新建代码配置 .....	73
2.12.3 修改代码配置 .....	74
2.12.4 删除代码配置 .....	74

2.13 服务管理 .....	75
2.13.1 查看服务管理模块 .....	75
2.13.2 创建推理服务 .....	76
2.13.3 编辑推理服务 .....	77
2.13.4 查看服务详情 .....	78
2.13.5 新增服务版本 .....	78
2.13.6 更新服务版本 .....	79
2.13.7 服务启停 .....	80
2.13.8 服务详情 .....	81
2.13.9 服务版本对比 .....	83
2.13.10 删除服务版本 .....	84
2.13.11 删除服务 .....	84
2.14 领域知识图谱 .....	85
2.14.1 查看领域知识图谱模块 .....	85
2.14.2 新增知识图谱图空间 .....	86
2.14.3 导入知识图谱数据 .....	88
2.14.4 删除知识图谱 .....	91
2.14.5 知识图谱控制台 .....	91
2.14.6 智能模板配置 .....	92
2.14.7 知识问答推理 .....	93

# 1 引言

## 1.1 文档范围和读者

本手册内容包括材料实验数据的自动分析平台功能支撑模块介绍,系统各模块的使用说明,工具使用说明。面向材料实验数据的自动分析平台的使用者,以及前端浏览用户的使用人员。为用户提供操作指导,详细向用户介绍了本系统具体的操作步骤和注意事项,使用户能够按照本手册进行准确无误的操作,保证系统的正常运行。

## 1.2 约定和术语

无特殊说明,平台指本软件系统网站。

## 2 材料实验数据的自动分析平台功能支撑模块

### 2.1 平台简介

平台提供了基于国产硬件的大模型训练优化框架、材料领域预训练数据及任务指令知识构建、训练优化及工具集成、协同开发服务环境、材料大模型推理及部署的全生命周期管理服务，为材料大模型训练与持续演化提供支撑。平台提供丰富的材料大模型训练组件、镜像、数据集、模型等，支持主流的深度学习框架及多种模型的开发方式。同时支持资源灵活调度、团队协作开发、版本管理等功能。

### 2.2 登录

介绍用户如何登录软件账户。

#### 步骤：

1.浏览器输入【<http://部署服务器 IP:31080/>】按 Enter 键进入登录界面，如图 2-1 所示。



图 2-1 登录界面

2. 登录界面输入用户名和密码，点击【登录】按钮
3. 如果用户名和密码正确，将登录成功页面跳转到平台工作空间，如图 2-2 所示。



图 2-2 工作空间

工作空间中介绍了智能材料科研平台用户的实验概览、多形态资源库的资源数量统计、算力积分使用情况，以及快速工作流程。

## 2.3 数据智能标注

数据标注平台是一个为机器学习和人工智能模型训练提供数据标注服务的工具，它能够帮助用户高效、准确地标注数据，并提供了一系列功能来管理、控制标注质量、协作和定制化，从而提高模型训练的效率 and 准确性。

### 2.3.1 查看数据智能标注模块

介绍用户如何进入数据智能标注模块项目界面，界面显示已创建的标注项目卡片。

**步骤：**

点击菜单栏【数据标注】菜单进入数据智能标注项目界面，如图 2-3 所示。

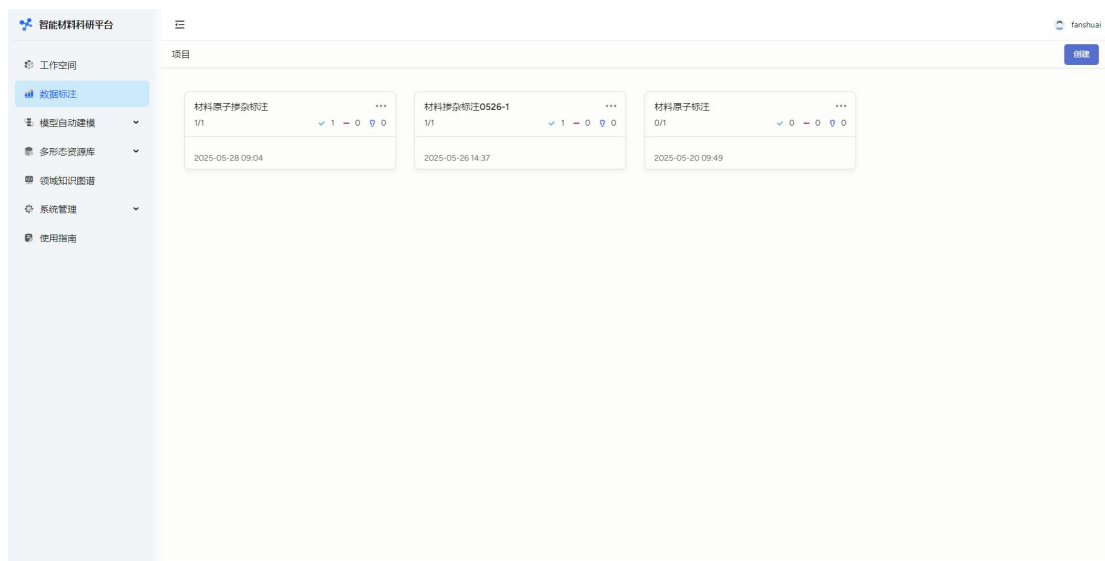


图 2-3 数据智能标注模块

### 2.3.2 创建标注项目

用户创建一个新的标注项目，将原始数据集导入到数据标注平台中，定义标注规则、标注类型（如分类、目标检测、语义分割等），准备进行标注。数据集可以是图像、文本、视频等多种类型。

#### 步骤：

1. 点击【创建】按钮弹出【创建项目】弹窗，默认位于【项目名称】界面，如图 2-4 所示，可自定义输入【项目名称】和【描述】。
2. 点击弹窗顶栏【数据导入】进入数据导入界面，如图 2-5 所示，可输入数据集的 URL 点击【添加 URL】按钮，或点击【上传文件】按钮选择上传文件，或拖放上传文件到弹窗界面即可成功上传标注文件。
3. 点击弹窗顶栏【标注配置】进入标注配置界面，如图 2-6 所示，点击选择项目的标注模板，并可修改模板配置，如图 2-7 所示。
4. 项目名称、数据导入和标注配置完成后，点击【保存】按钮成功创建标注项目，项目页面显示创建的标注项目卡片。

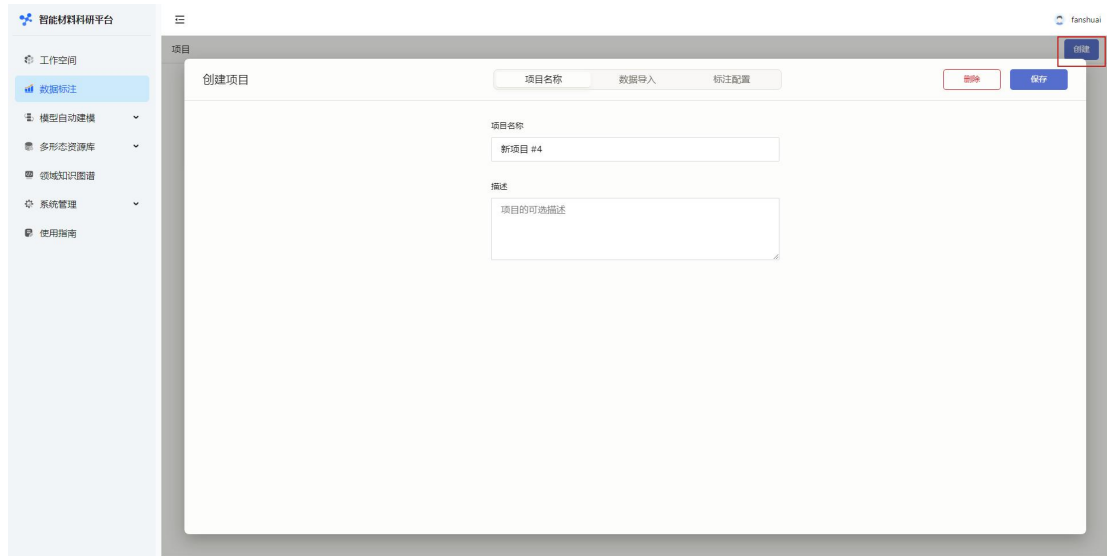


图 2-4 项目名称

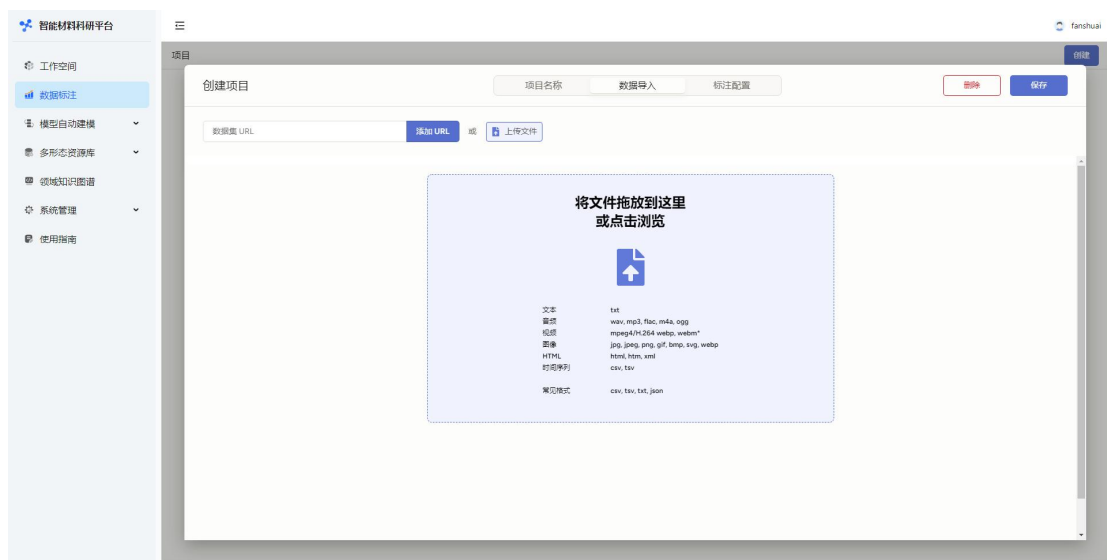


图 2-5 数据导入

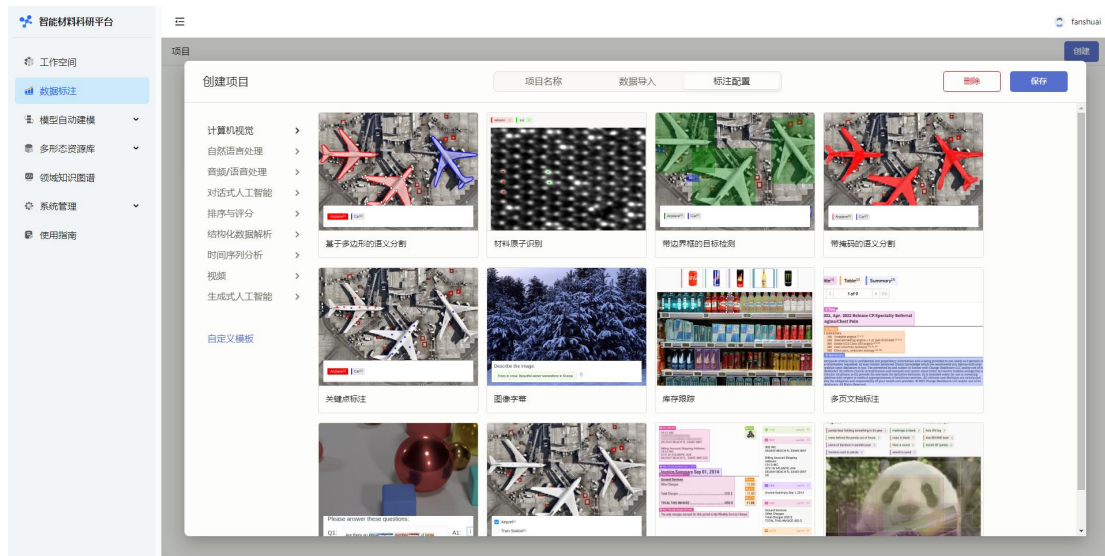


图 2-6 标注配置

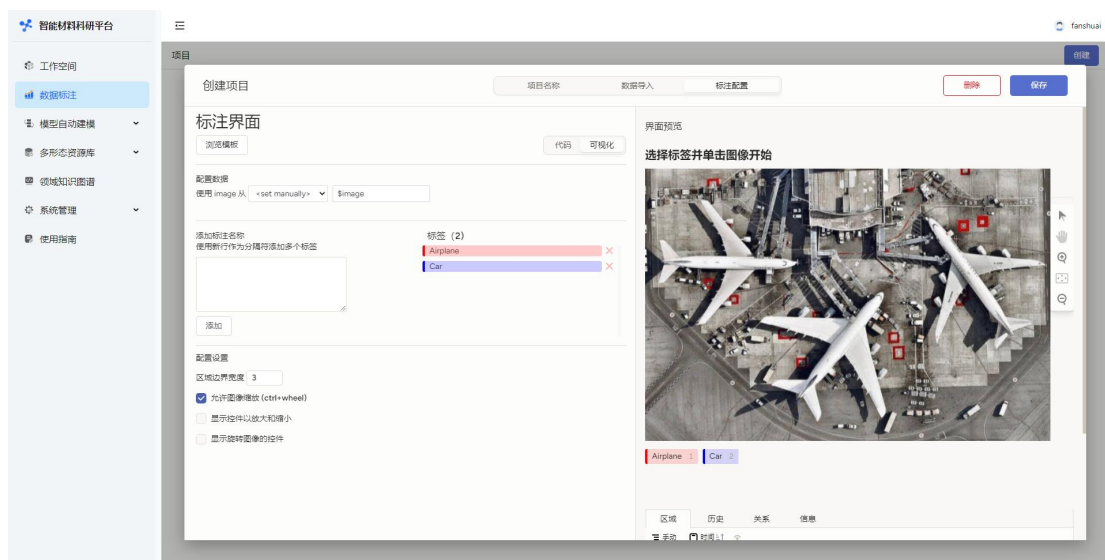


图 2-7 标注模板配置

### 2.3.3 标注数据

用户对标注任务进行数据标注。

**步骤：**

1. 点击标注项目卡片进入项目标注任务列表，如图 2-8 所示。
2. 点击【标注所有任务】进入标注任务对数据进行标注，如图 2-9 所示。
3. 标注完成点击【提交】按钮保存标注成功。

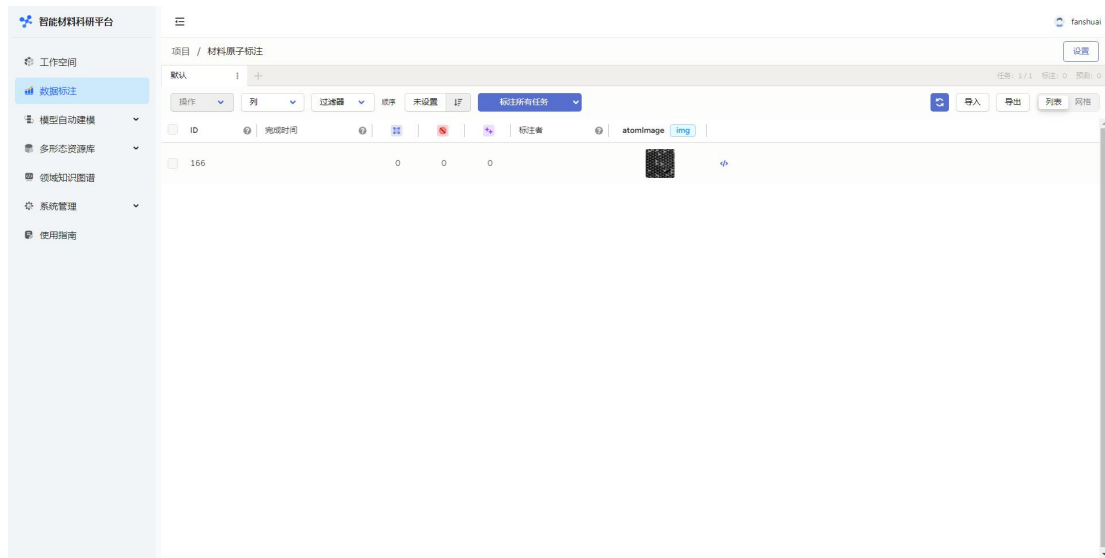


图 2-8 标注任务列表

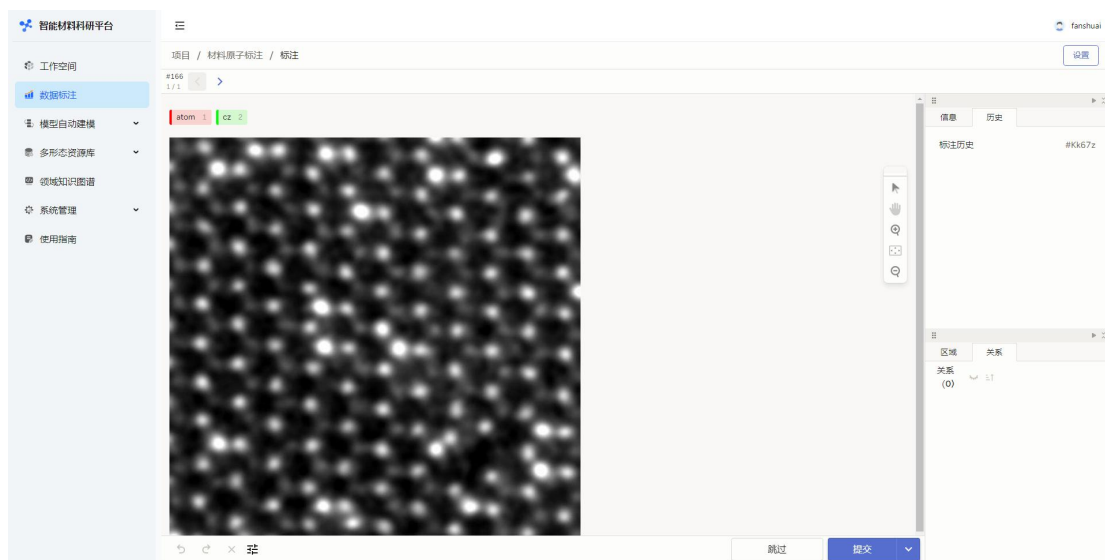


图 2-9 标注数据

### 2.3.4 标注结果导出

标注任务完成后，用户可以将标注结果导出到本地或者平台数据集，用于后续的模式训练。

**步骤：**

1. 点击【导出】按钮弹出【导出数据】弹窗，如图 2-10 所示，可选择数据集导出格式。
2. 如果点击弹窗【导出】按钮则下载到本地。
3. 如果点击弹窗【导出到平台】按钮，则弹出【导出数据到平台】弹窗，如图 2-11 所示，选择数据集，输入数据集版本和版本描述，点击【确定】按钮

则导出到平台数据集供模型训练使用。

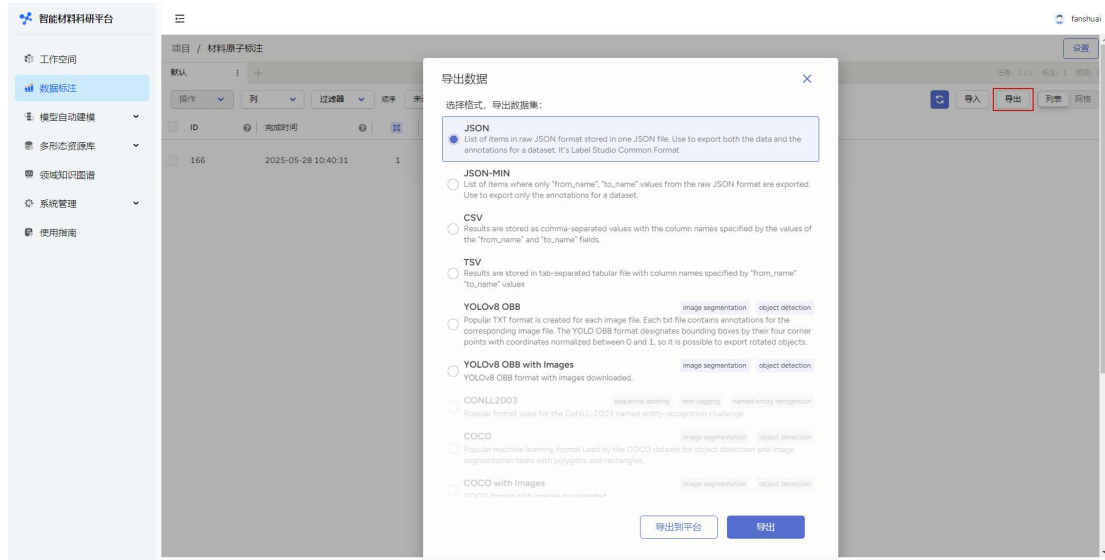


图 2-10 导出数据

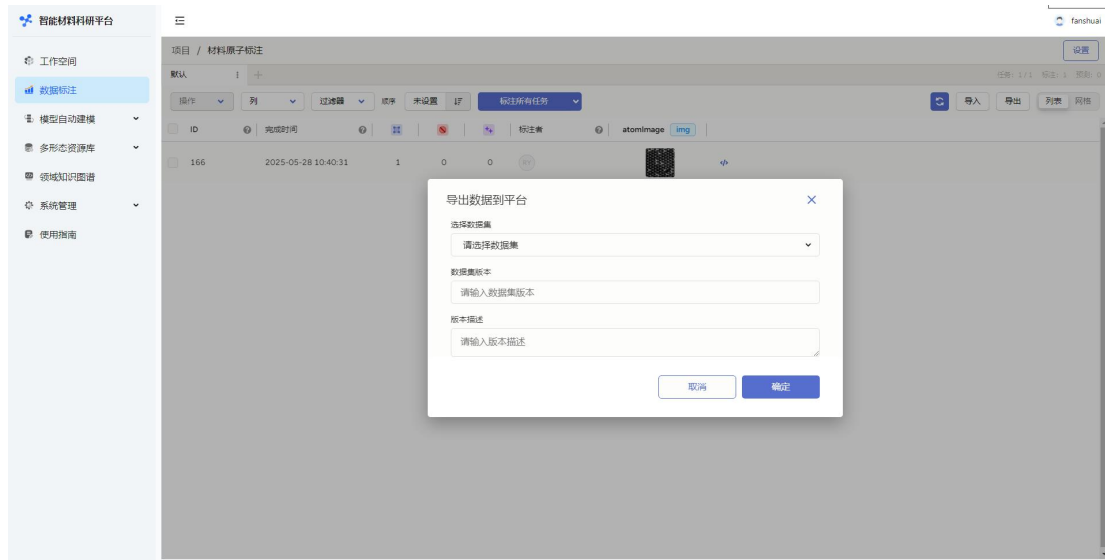


图 2-11 导出数据到平台

### 2.3.5 标注任务查询

查询已标注的数据，查看标注详情或统计信息。

步骤：

1. 标注任务列表界面点击【过滤器】-【添加过滤器】按钮，如图 2-12 所示。
2. 设置过滤条件可对项目标注任务过滤查询。

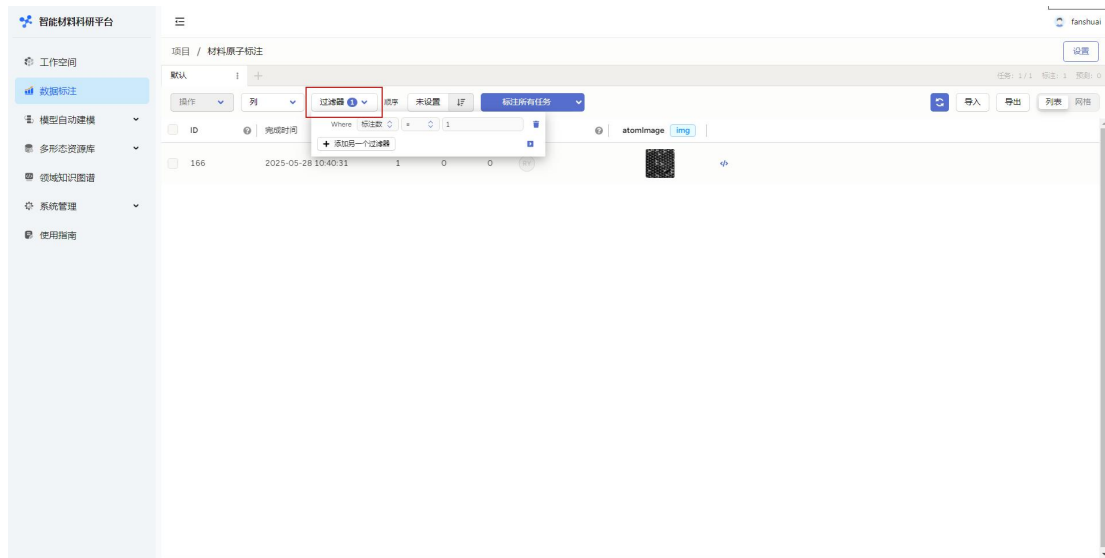


图 2-12 添加过滤器

### 2.3.6 删除标注项目

删除不再需要的标注项目，释放存储空间。同时系统会确保删除操作的安全性，删除前进行提示，避免误删重要项目。

步骤：

1. 点击【设置】按钮进入项目设置界面。
2. 点击菜单【危险区域】-【删除项目】按钮弹出【操作确认】弹框，如图 2-13 所示。
3. 点击弹框【确认】按钮成功删除标注项目。



图 2-13 标注项目删除

## 2.4 流水线模板

流水线模板为用户提供了数据处理、模型训练、模型测试、数据导出等丰富的组件，用户可以根据业务场景的不同，通过可视化的方式快速构建 MLOps 流水线，并对每个组件的参数进行个性化设置。这样，模型训练过程不仅得到自动化，还能适应不同的训练需求和环境，确保模型开发的高效性和一致性。

### 2.4.1 查看流水线模板模块

介绍用户如何进入流水线模板列表界面，可通过关键词快速检索目标流水线模板。

步骤：

1. 点击菜单栏【模型自动建模】-【流水线模板】菜单进入流水线模板列表界面，流水线模板列表包括序号、流水线名称、流水线描述、创建时间、修改时间和操作，如图 2-20 所示。

2. 流水线模板列表上方筛选框输入关键字，按 Enter 键将进行模糊搜索，列表只显示流水线名称包括关键字的流水线模板，如图 2-21 所示。

序号	流水线模板名称	流水线模板描述	创建时间	修改时间	操作
1	原子位置识别模型	对原子位置识别的模型进行测试	2025-05-19 16:03:39	2025-05-28 10:07:46	编辑 复制 删除
2	主动学习模型	对主动学习的模型进行测试	2025-05-20 13:52:05	2025-05-28 10:07:12	编辑 复制 删除
3	材料是否可降解模型	测试自动机器学习训练的模型的效果	2025-05-19 15:41:53	2025-05-28 10:07:02	编辑 复制 删除
4	THUCNews文本分类流水线	THUCNews文本分类流水线	2025-05-27 09:20:14	2025-05-28 10:06:36	编辑 复制 删除
5	视频分类流水线	视频分类流水线	2025-05-23 10:41:18	2025-05-28 09:57:27	编辑 复制 删除
6	数据预处理	包含数据处理和数据导出	2025-05-22 11:04:27	2025-05-27 17:16:22	编辑 复制 删除
7	手写体识别分类	对手写体识别分类主动学习的模型进行测试	2025-05-27 10:38:44	2025-05-27 10:39:07	编辑 复制 删除
8	原子掺杂识别模型训练	包含模型训练、模型测试、模型保存	2025-05-14 16:19:00	2025-05-26 11:47:03	编辑 复制 删除
9	介电材料模型训练	包含模型训练、模型导出	2025-05-22 17:03:53	2025-05-22 17:18:22	编辑 复制 删除

图 2-20 流水线模板模块

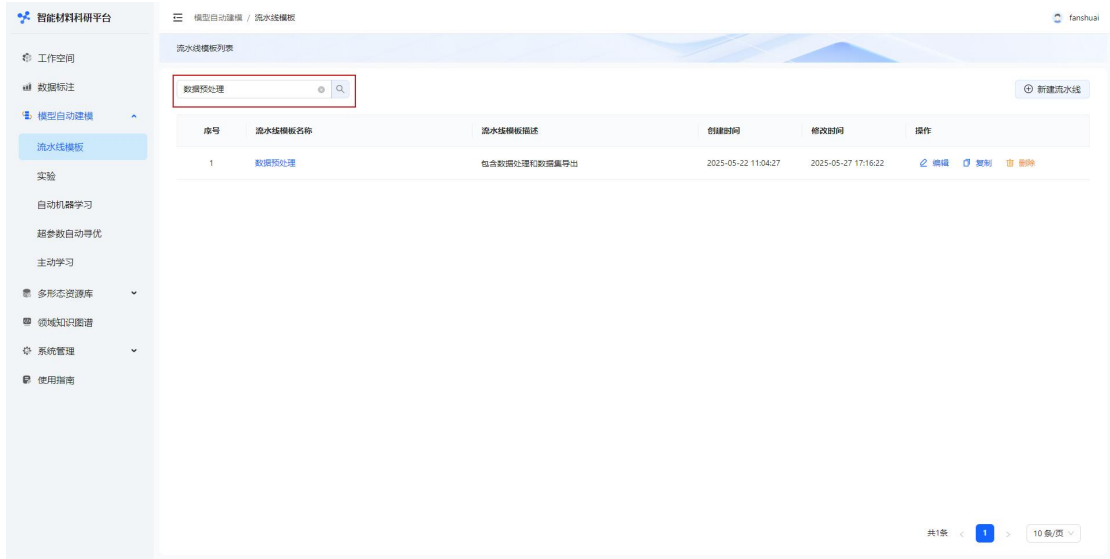


图 2-21 查询流水线

### 2.4.2 新建流水线

创建新的流水线模板，定义名称和描述，便于后续复用和共享。

步骤：

1. 点击【新建流水线】按钮弹出【新建流水线】弹窗，如图 2-22 所示。
2. 输入流水线名称和流水线描述，点击弹窗【确定】按钮成功新建流水线。

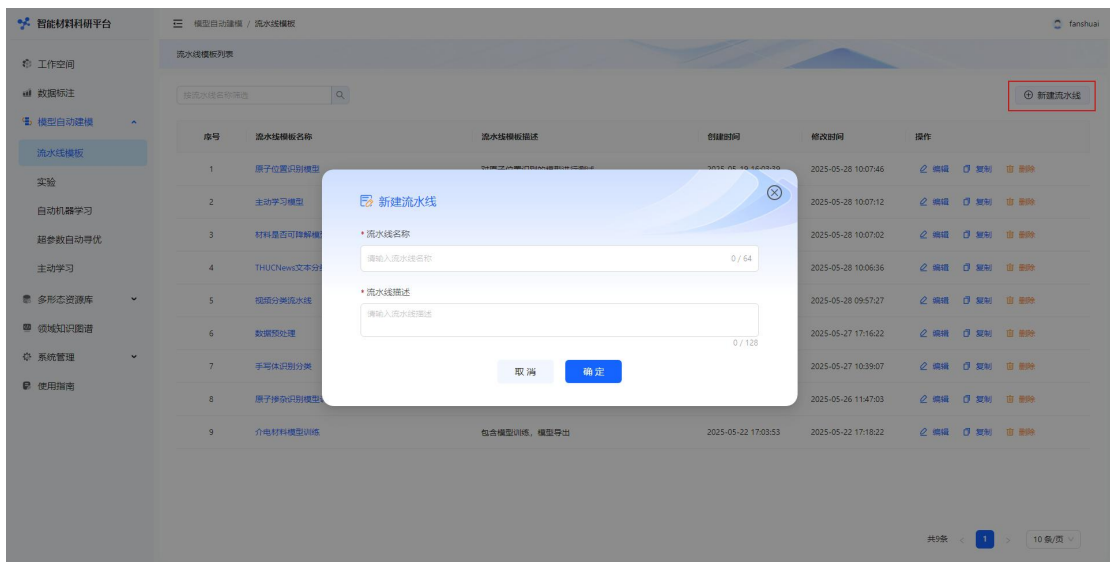


图 2-22 新建流水线

### 2.4.3 配置流水线

新建流水线完成后，提供可视化的拖拽式界面，用户可以通过图形化方式灵活编排流水线组件，定义任务流程和数据流向，降低使用门槛。

步骤：

1. 在流水线列表中点击【流水线名称】，可以进入流水线详情界面。
  2. 根据业务场景从左侧组件库中拖动组件到画布中，然后根据业务流程连接各个组件，如图 2-23 所示。
  3. 画布双击组件，弹出【编辑任务】侧边栏可以配置组件的相关参数，如图 2-24 所示。
  4. 点击【全局参数】弹出全局参数侧边栏，点击【+ 流水线参数】按钮可新增全局参数设置，如图 2-25 所示。
- 设置成功的流水线全局参数可点击【参数】-【全局参数】选择参数，输入框内显示 $\{\{\text{参数名称}\}\}$ 引用，如图 2-26 所示。
5. 流水线中通常需要引用其他组件任务的输出结果，可点击【参数】-【组件任务】选择参数，输入框内显示 $\{\{\text{task\_id.param\_name}\}\}$ 引用，如图 2-27 所示，其中“git-clone-c0724278”是代码拉取的 task\_id，--code\_output 是代码拉取组件的输出参数名称。

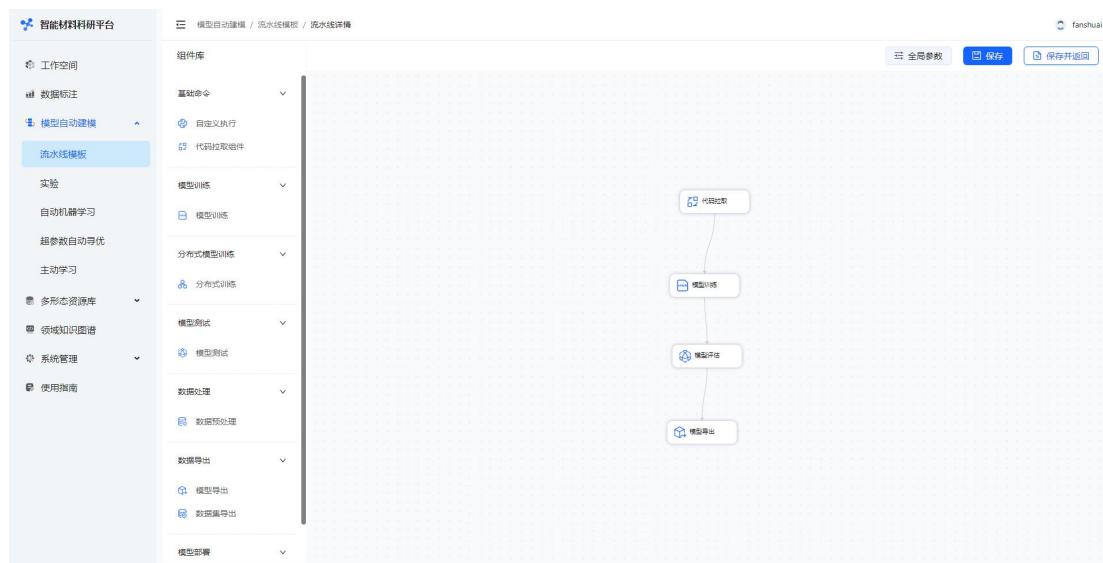


图 2-23 流水线拓扑配置

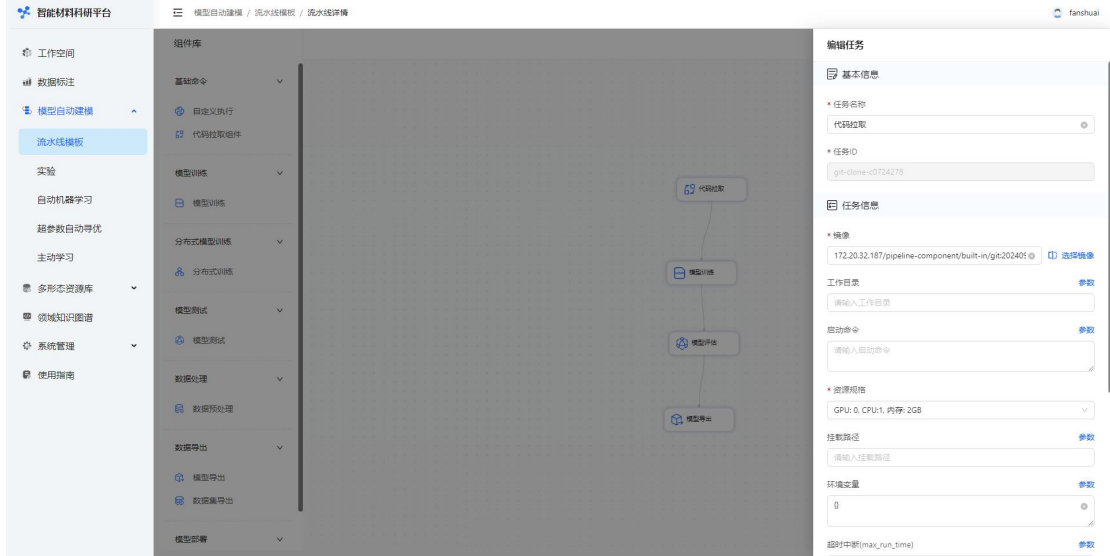


图 2-24 编辑任务

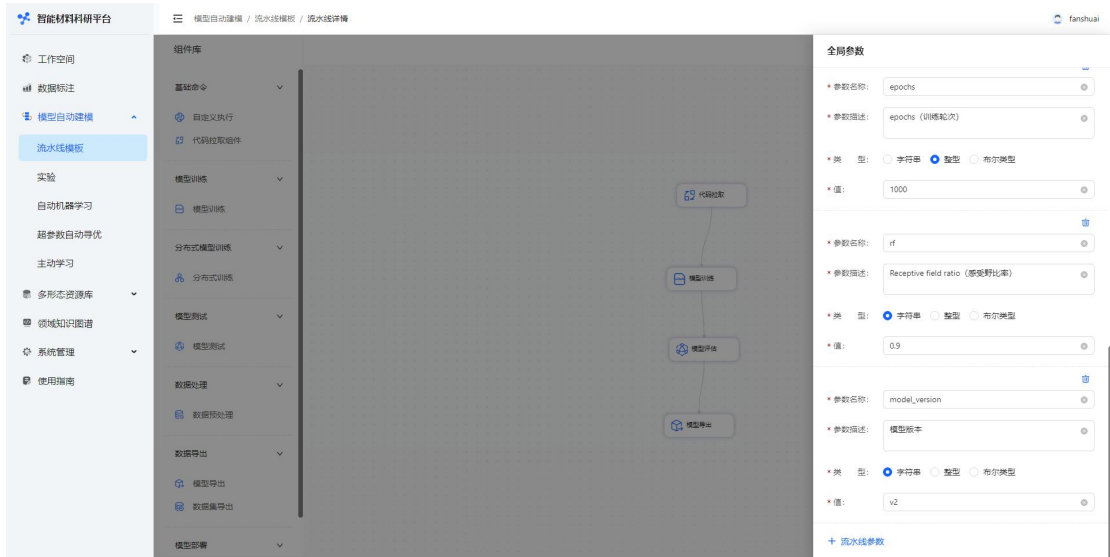


图 2-25 全局参数

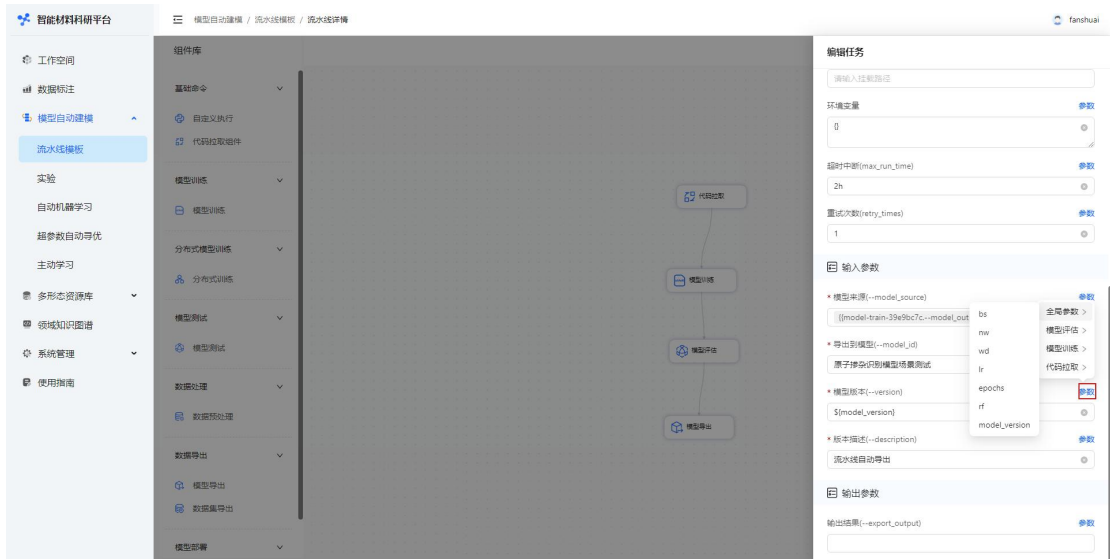


图 2-26 全局参数引用

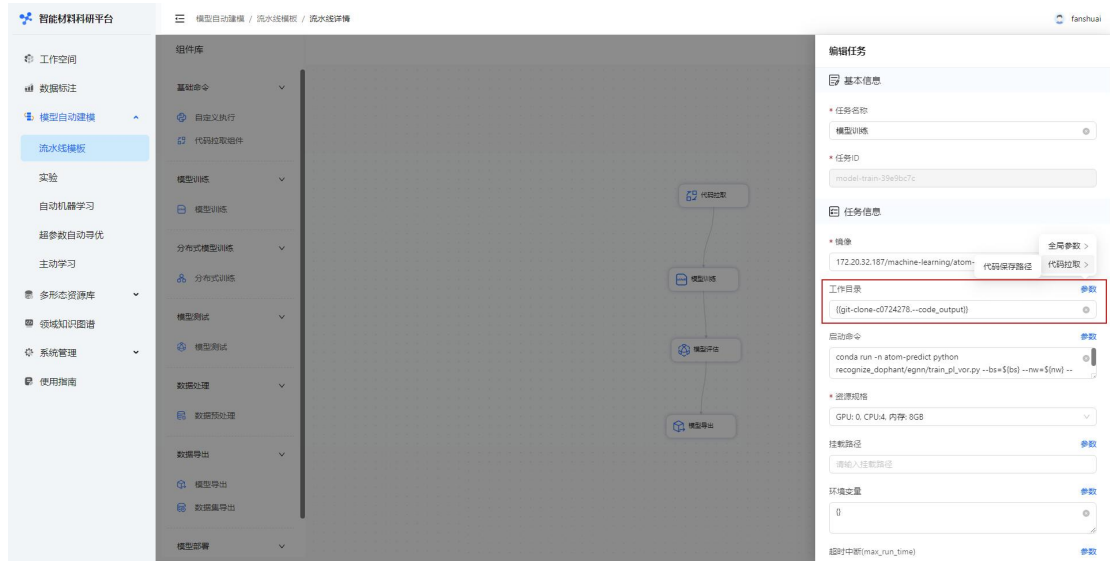


图 2-27 组件产物输出引用

## 2.4.4 组件介绍

### 常用组件介绍

**代码拉取：**拉取组件运行所需要的代码，目前支持拉取 git 代码，不支持拉取其他方式的代码（svn 等），支持公有和私有 git 仓库的拉取。

**模型训练：**提供开箱即用的深度学习训练环境，支持 PyTorch、TensorFlow、Keras 等主流框架，用户可直接调用平台托管的精选数据集和预训练模型进行训练，大幅降低机器学习入门门槛。

**模型测试：**支持 PyTorch、TensorFlow、Keras 等主流框架的模型测试，用户可直接使用平台托管的数据集和模型进行测试，同时支持测试结果的持久化和可视化。

**数据预处理：**通用数据预处理，用户可以对选择的数据集进行处理，如果处理后的数据需要保存到数据集中，可以和数据集导出组件配合使用。

**模型导出：**将模型训练输出的模型导出平台保存以便后续使用。如果没有注册过模型，需要先注册模型才能导出。

**数据集导出：**将数据处理后的结果导出到平台保存以便后续使用，如果数据集不存在，需要先创建数据集才能导出。

组件参数分为“基本信息”、“任务信息”、“输入参数”，“输出参数”

四个部分，如图 2-28 所示。

**基本信息：**包含任务名称（在画布中显示的名称）、任务 id（组件从组件库侧边栏拖动到画布中自动生成，格式为“组件名称-随机字符串”）。

**任务信息：**包含任务执行的通用信息。

**镜像：**任务运行需要的镜像，从平台提供的镜像列表中选择。

**工作目录：**启动命令运行的工作目录。

**启动命令：**如果启动命令为空，则使用镜像默认的启动命令，如果启动命令不为空，则会覆盖镜像的默认启动命令。启动命令后面可以带命令需要执行的参数。

**资源规格：**任务运行需要的资源，目前支持 CPU/GPU 类型的规格，后续会支持其他的资源类型。

**环境变量：**任务运行时需要注入的环境变量，格式为“key=value”，每行设置一个环境变量，多个环境变量需要换行设置。

**输入参数：**输入参数分为两类，一类是挂载参数，一类是普通参数。所有的输入参数都会以参数的方式传给任务的启动命令，传入的参数名称为输入参数括号内的字符串。

**输出参数：**所有的输出参数都会以参数的方式传给任务的启动命令。输出参数都会被挂载到任务执行的容器中，用户将结果写入挂载的路径中结果会持久化。

以模型训练为例，启动命令为“python train.py --batch\_size=128 --epoch=2”，任务执行时真正的启动命令是“python train.py --batch\_size=128 --epoch=2 --dataset=/{{workflow-name}}/dataset --model\_name=/{{workflow-name}}/model --model\_output=/model”

说明：

{{workflow-name}}是个变量，每次运行都会自动生成不同的实例名字

输入参数：

--dataset=/workflow-name/dataset 数据集挂载的路径，用户从--dataset 参数中可以读取选择的数据集，dataset 里面的 zip 是只读挂载的，需要用户把 zip 解压到其他的可读写目录。

`--model_name=/workflow-name/model` 模型挂载的路径，用户从 `--model_name` 参数中可以读取选择的模型，注意 `--model_name` 参数是模型的存放路径，模型文件需要用户从 `--model_name` 中获取模型文件。

输出参数：

`--model_output=/model` 模型的输出路径，输出到该路径的模型会被持久化，任务结束可以查看和下载。

### 编辑任务

#### 基本信息

\* 任务名称

模型训练

\* 任务ID

model-train-39e9bc7c

#### 任务信息

\* 镜像

172.20.32.187/machine-learning/atom-egnn.v2 选择镜像

工作目录 参数

{{git-clone-c0724278.--code\_output}}

启动命令 参数

```
conda run -n atom-predict python
recognize_dophant/egnn/train_pl_vor.py --bs=${bs} --nw=${nw} --
```

\* 资源规格

GPU: 0, CPU:4, 内存: 8GB

挂载路径 参数

请输入挂载路径

环境变量 参数

\$

超时中断(max\_run\_time) 参数

2h

重试次数(retry\_times) 参数

1

#### 输入参数

\* 选择数据集(--dataset) 参数

原子掺杂识别场景测试.v1 选择数据集

选择模型(--model\_name) 参数

原子掺杂识别模型.v1 选择模型

#### 输出参数

\* 模型输出路径(--model\_output) 参数

/model

图 2-28 组件参数

## 2.4.5 编辑流水线

对已有流水线模板进行编辑，修改名称、描述等。

**步骤：**

1. 流水线模板列表点击【编辑】按钮弹出【编辑流水线】弹窗，如图 2-29 所示。
2. 可编辑流水线名称或流水线描述，点击弹窗【确定】按钮编辑流水线保存成功。



图 2-29 编辑流水线

## 2.4.6 复制流水线

通过复制现有流水线模板快速创建新模板，保留原有配置，提高工作效率。

**步骤：**

- 流水线模板列表点击【复制】-【确认】按钮能快速复制出一条流水线，复制出的流水线名称为“原流水线名称-copy-随机字符串”，如图 2-30 所示。

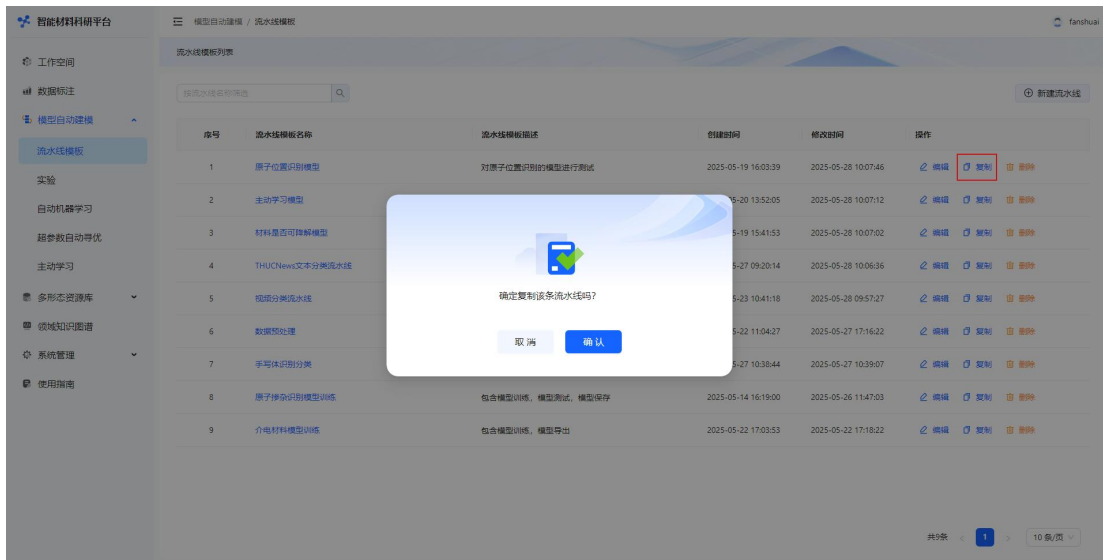


图 2-30 复制流水线

### 2.4.7 删除流水线

选择不再需要的流水线模板进行删除操作，释放存储空间，同时系统会确保删除操作的安全性，当流水线已有对应的运行实验，则不可删除，删除时会提示，避免误删重要模板。

#### 步骤：

1. 流水线模板列表点击【删除】-【确认】按钮，如图 2-31 所示。如果流水线模板无实验关联，则成功删除流水线模板，如果存在关联实验则删除失败且弹框提示。
2. 点击菜单【模型自动建模】-【实验】进入实验模块，实验列表中删除流水线模板关联的实验，参考章节 2.6.8。
3. 返回流水线模板列表再次点击【删除】-【确认】按钮，成功删除流水线模板。

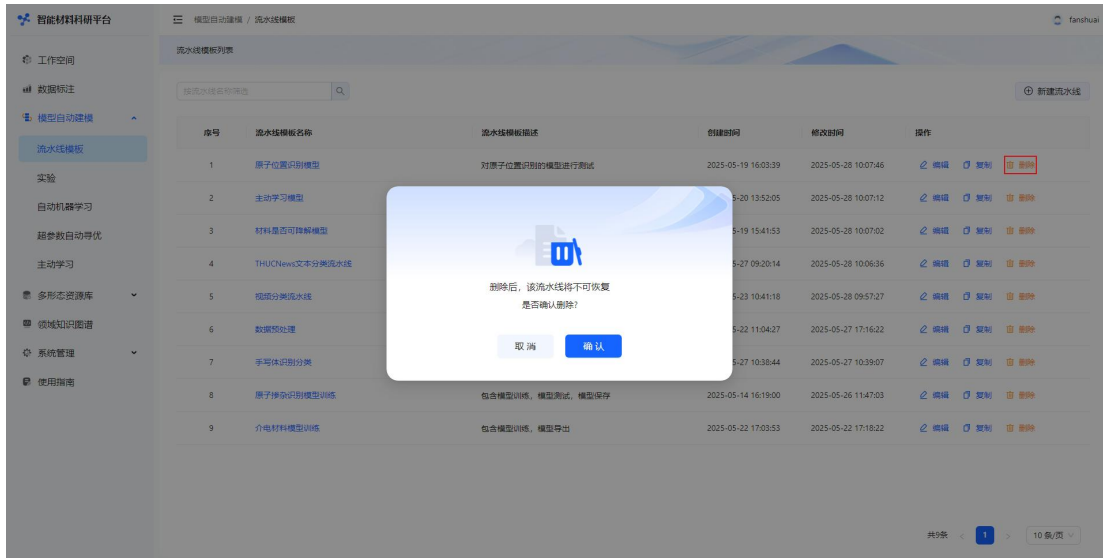


图 2-31 删除流水线

## 2.5 实验

流水线模板配置完成后, 用户可以选择流水线, 动态调参进行实验。

### 2.5.1 查看实验模块

介绍用户如何进入实验列表界面, 可通过关键词快速检索目标实验。

**步骤:**

1. 点击菜单栏【模型自动建模】-【实验】菜单进入实验列表界面, 实验列表包括实验名称、关联流水线名称、实验描述、最近五次运行状态和操作, 如图 2-32 所示。

2. 实验列表上方筛选框输入关键字, 按 **Enter** 键将进行模糊搜索, 列表只显示实验名称包括关键字的实验, 如图 2-33 所示。

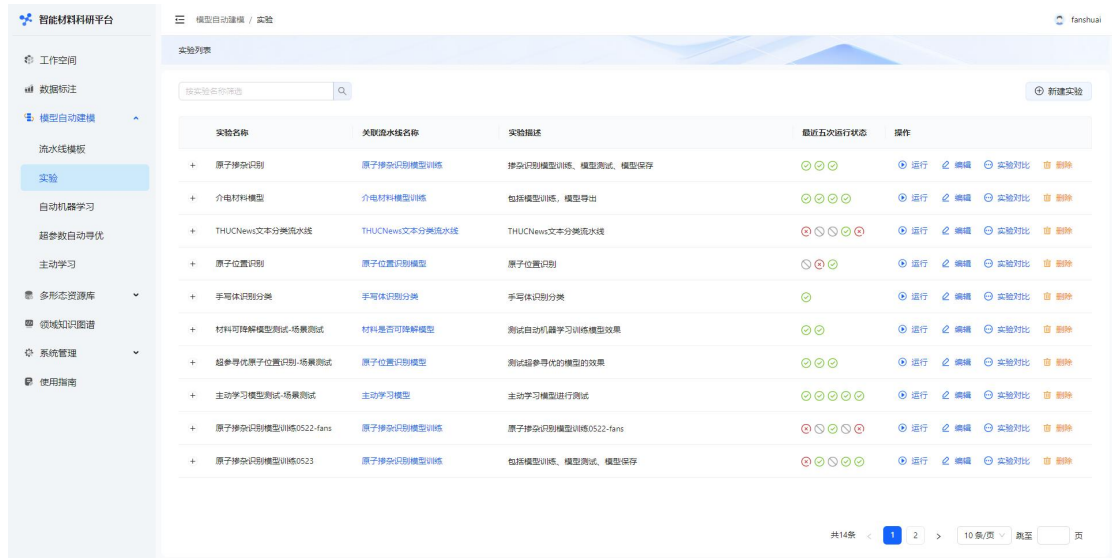


图 2-32 实验模块

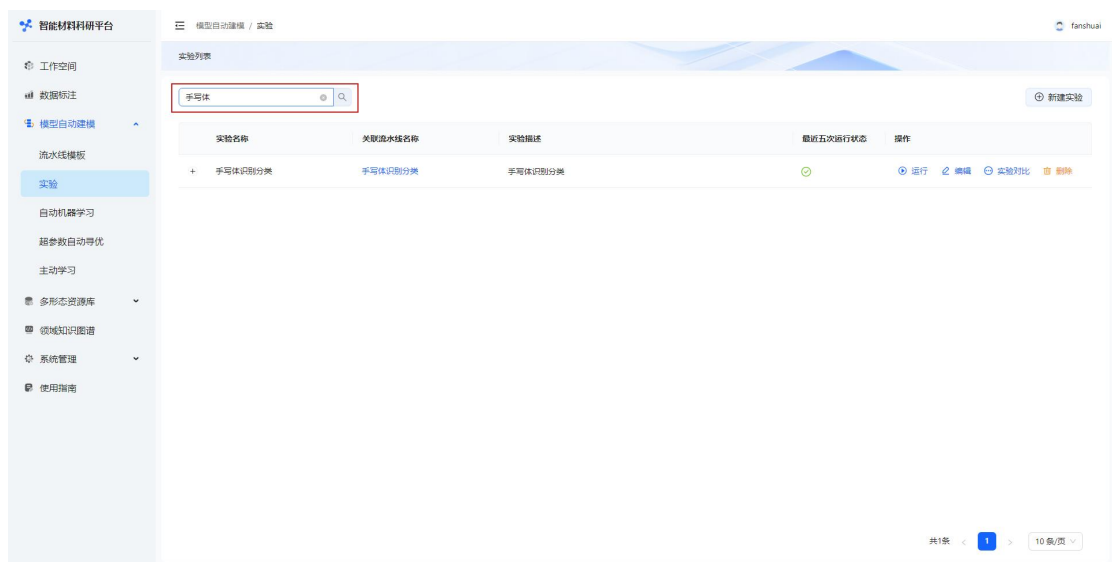


图 2-33 查询实验

## 2.5.2 新建实验

创建新的实验任务，绑定流水线模板。

步骤：

1. 点击【新建实验】按钮弹出【新建实验】弹窗，如图 2-34 所示。
2. 输入实验名称和实验描述，选择流水线，根据实际情况修改运行参数，点击弹窗【确定】按钮成功新建实验。

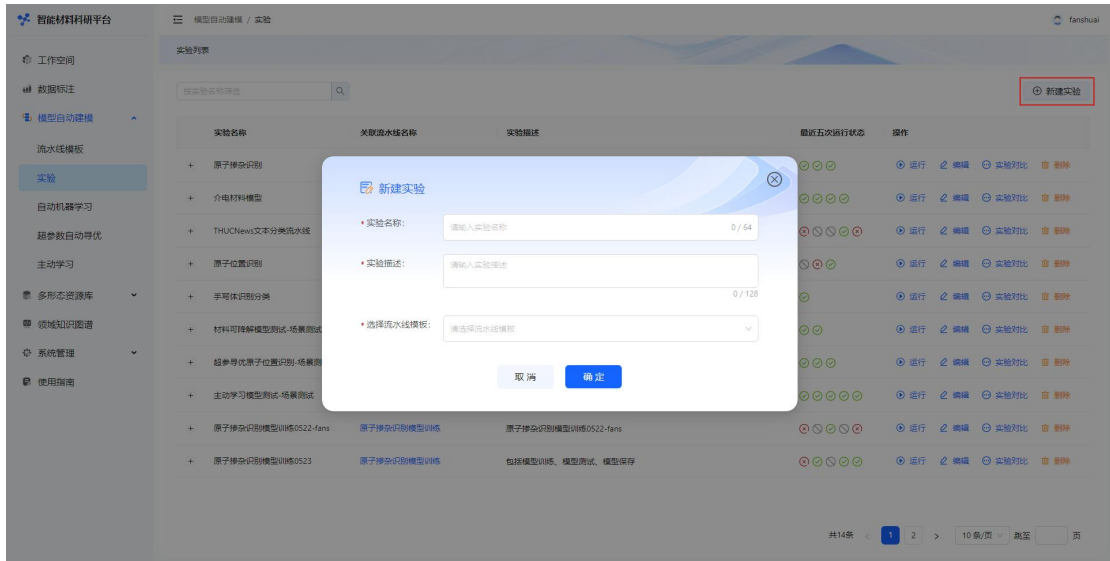


图 2-34 新建实验

### 2.5.3 编辑实验

对已有实验进行编辑和优化，调整全局参数。

**步骤：**

1. 实验列表点击【编辑】按钮弹出【配置实验】弹窗，如图 2-35 所示。
2. 可编辑实验名称、实验描述或运行参数值，点击弹窗【确定】按钮编辑实验保存成功。



图 2-35 编辑实验

### 2.5.4 运行实验

启动实验任务，生成实验实例，系统会自动分配资源并执行实验流程。

**步骤：**

实验列表点击【运行】按钮即可运行实验，展开实验实例列表自动生成一条实验实例，如图 2-36 所示。

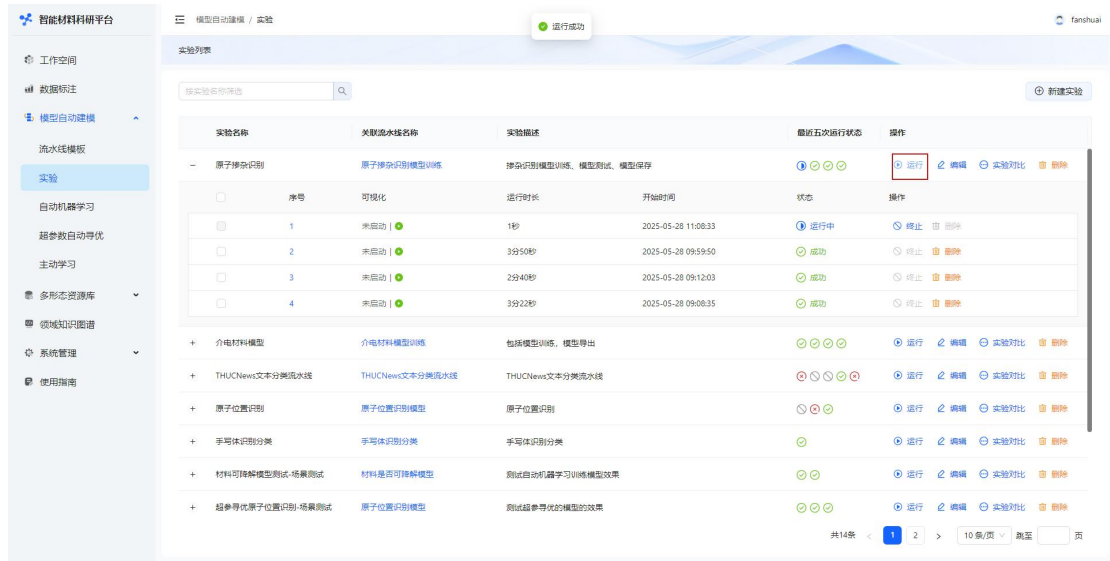


图 2-36 运行实验

### 2.5.5 实验详情查看

查看当前实验对应的实验实例列表，查看各个实例的状态。

**步骤：**

1. 实验运行完成后，点击实验实例列表的【序号】，可以进入实验实例查看实验运行的情况，如图 2-37 所示。
2. 点击【执行参数】，可以查看此次实验的执行参数配置情况，如图 2-38 所示。
3. 双击画布的组件弹出对应组件的【任务执行详情】侧边栏，包括运行状态、运行时间和日志详情，如图 2-39 所示。
4. 在【任务执行详情】侧边栏点击【配置参数】可以查看当前任务组件的配置参数，如图 2-40 所示。
5. 如果组件任务运行结束，点击【输出结果】可以查看和下载组件的输出结果，如图 2-41 所示。

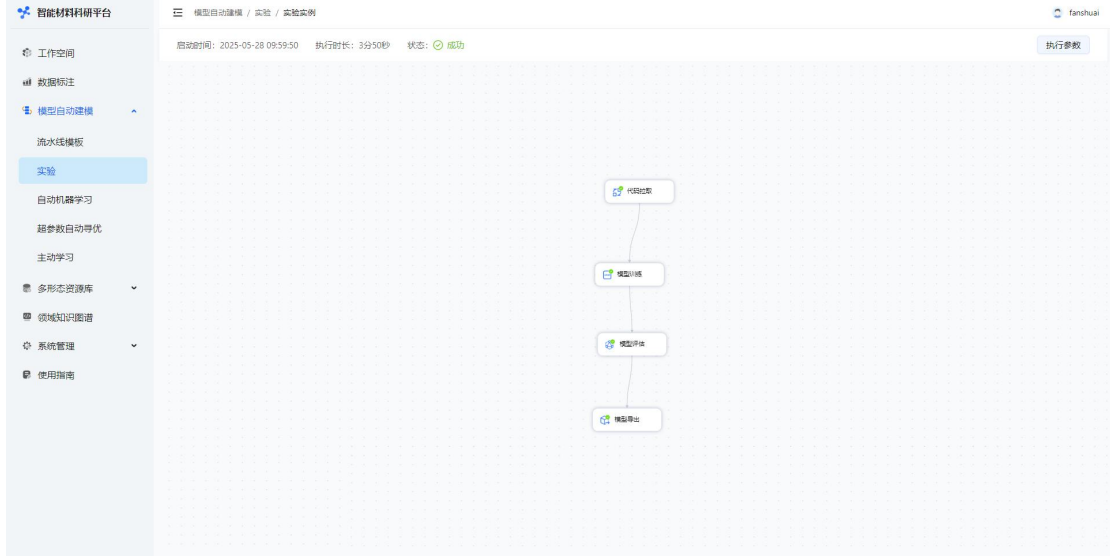


图 2-37 实验实例查看

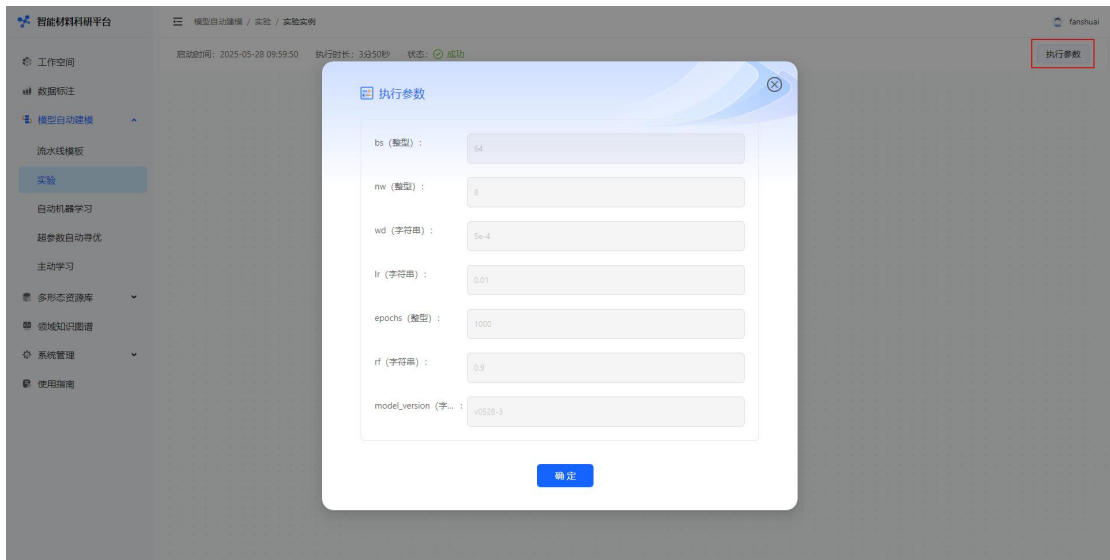


图 2-38 执行参数

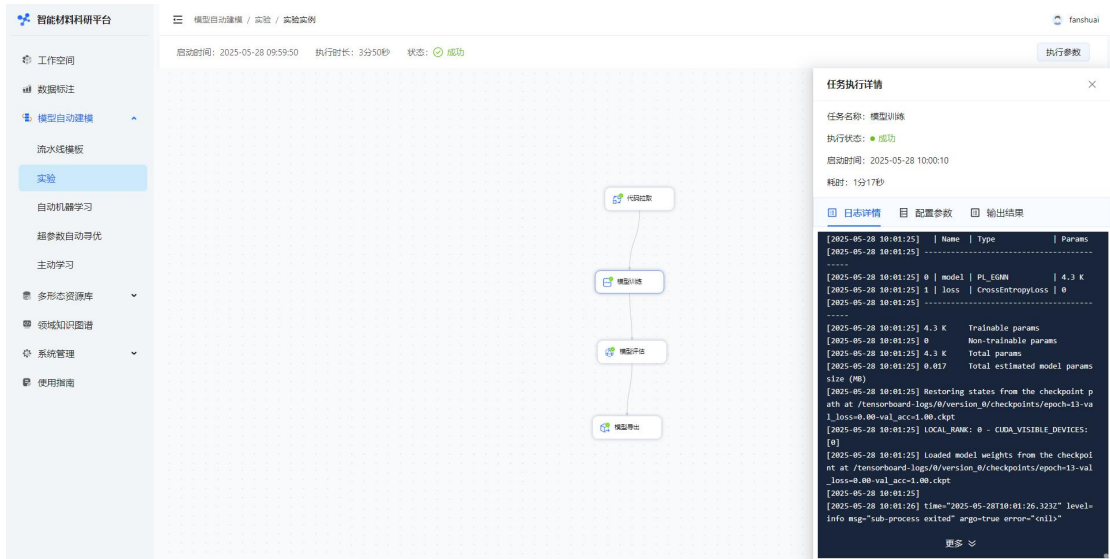


图 2-39 任务执行详情

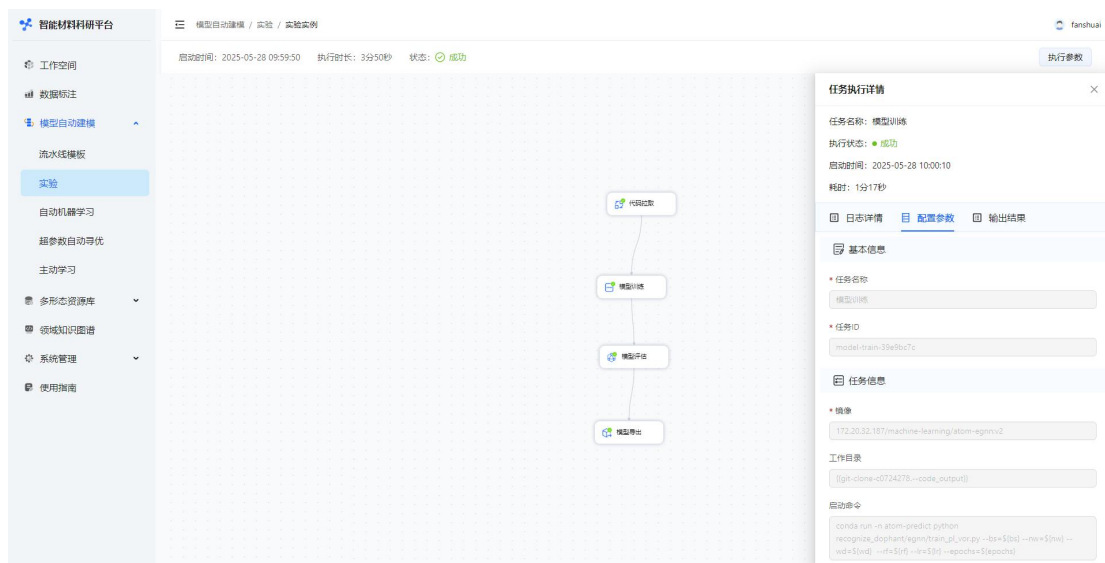


图 2-40 配置参数

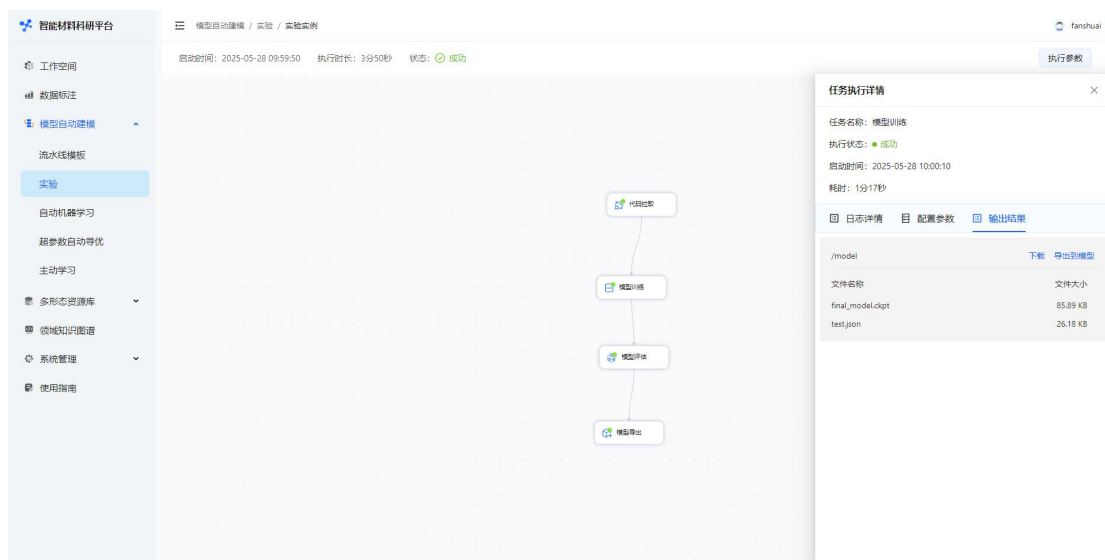


图 2-41 输出结果

### 2.5.6 实验过程参数变化查看

对于模型训练，有时候需要分析和查看模型训练过程中参数的变化，平台集成了 tensorboard 可视化面板查看参数的变化。模型训练和分布式训练组件默认开启 tensorboard 可视化记录。

**步骤：**

1. 实验实例列表点击绿色启动按钮，即可启动 tensorboard 可视化。
2. 启动成功后点击链接按钮，如图 2-43 所示，即可进入 tensorboard 面板分析和查看模型训练过程中参数的变化，如图 2-44 所示。

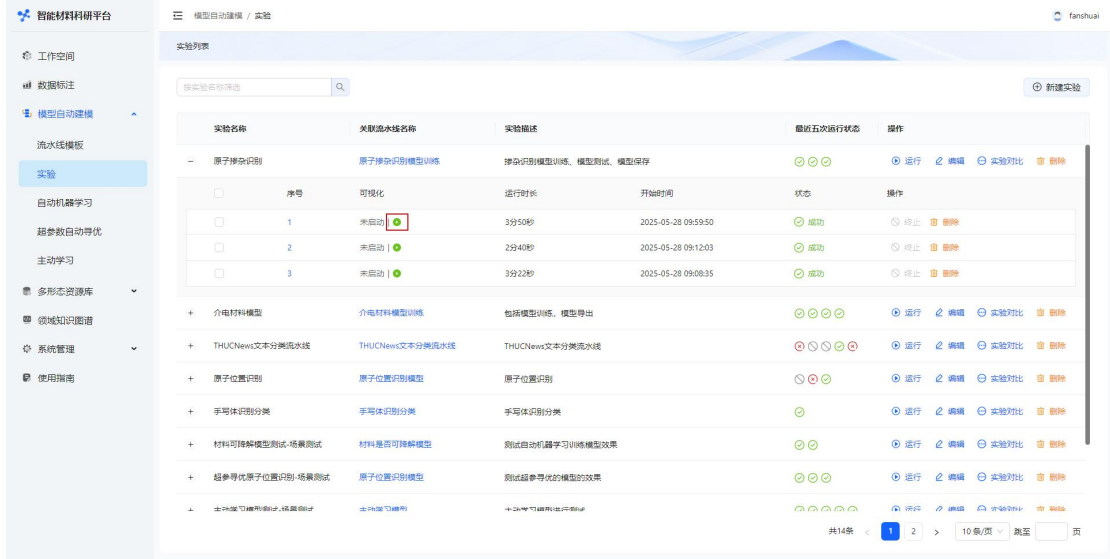


图 2-42 可视化启动按钮



图 2-43 链接按钮

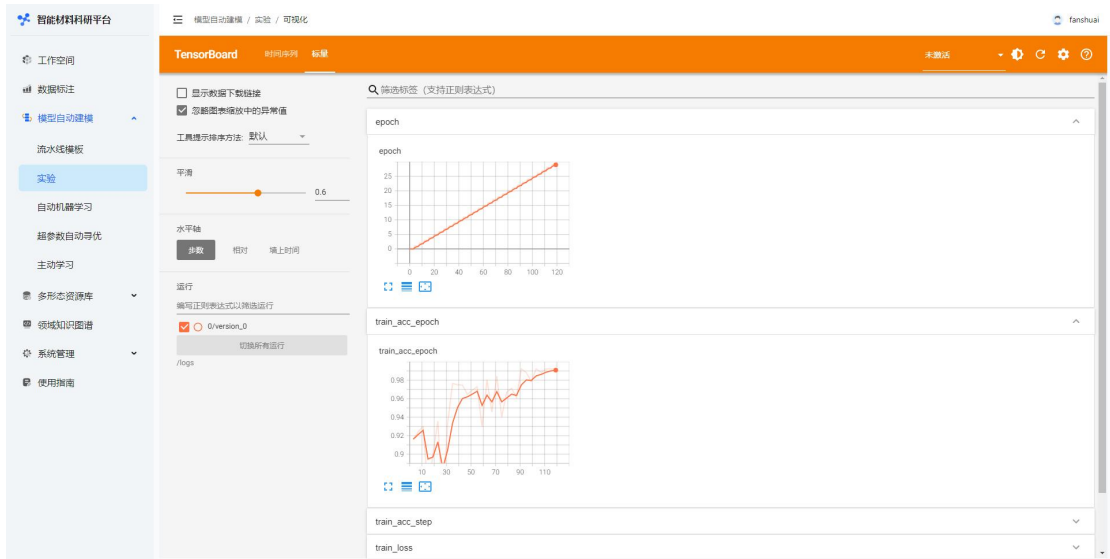


图 2-44 tensorboard 面板

### 2.5.7 终止实验

手动终止正在运行的实验实例，释放计算资源，适用于实验异常或需要中断的场景。

**步骤：**

实验实例列表点击【终止】-【确认】按钮，可成功终止正在运行的实验实例，如图 2-45 所示。

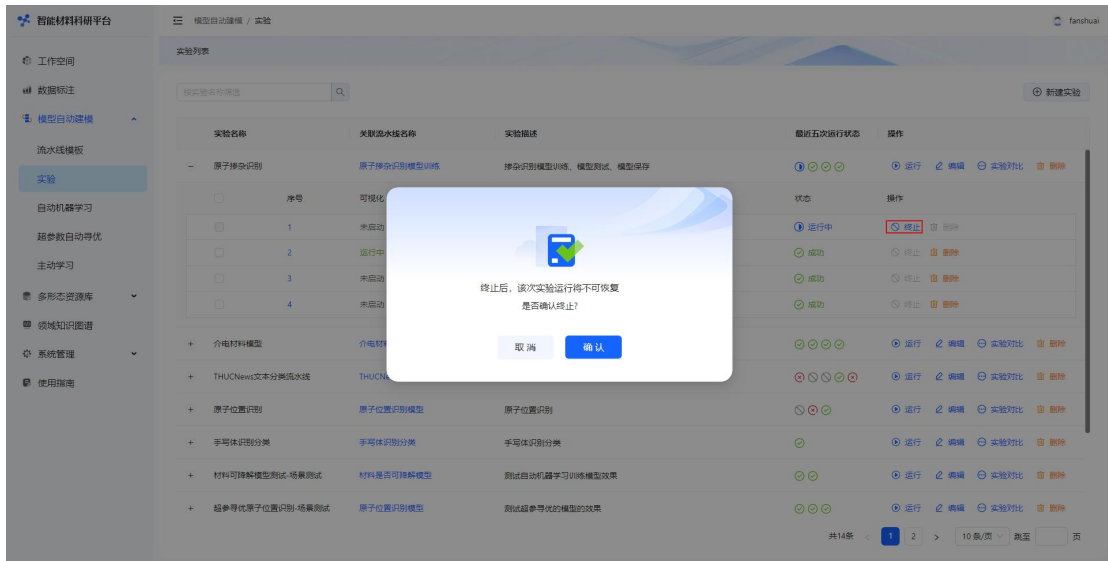


图 2-45 终止实验

### 2.5.8 删除实验

选择不再需要的实验进行删除操作，释放存储空间，同时系统会确保删除操作的安全性，当实验存在实验实例则无法删除，删除前进行提示，避免误删重要实验。

**步骤：**

1. 实验列表点击【删除】-【确认】按钮，如图 2-46 所示，如果实验无实验实例，则成功删除实验，如果存在实验实例则删除失败且弹框提示。
2. 点击【+】展开实验实例列表，勾选实验所有的实验实例，点击【删除】-【确认】按钮成功批量删除实验实例，如图 2-47 所示。
3. 实验列表再次点击【删除】-【确认】按钮，成功删除实验。

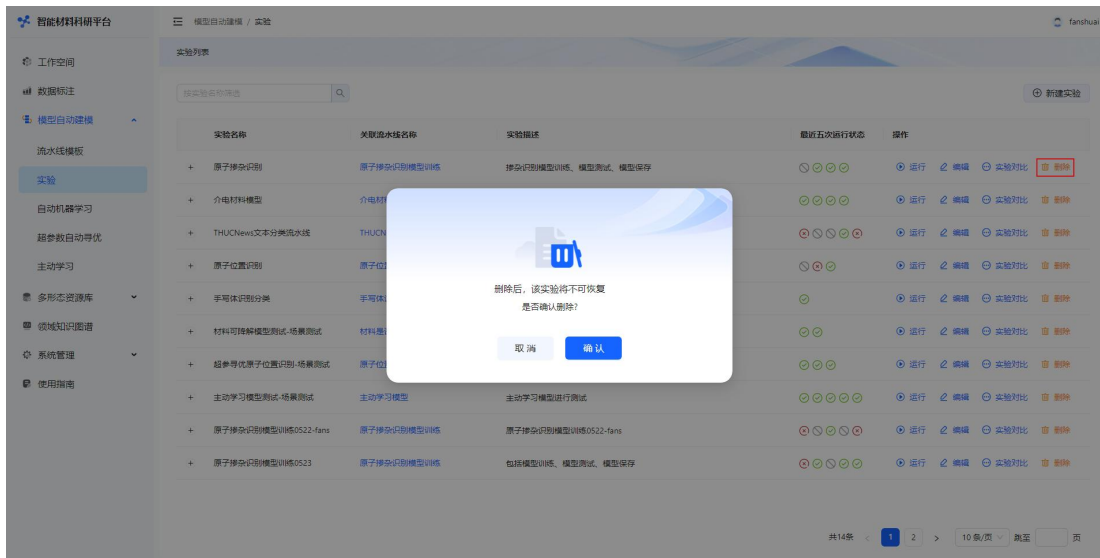


图 2-46 删除实验

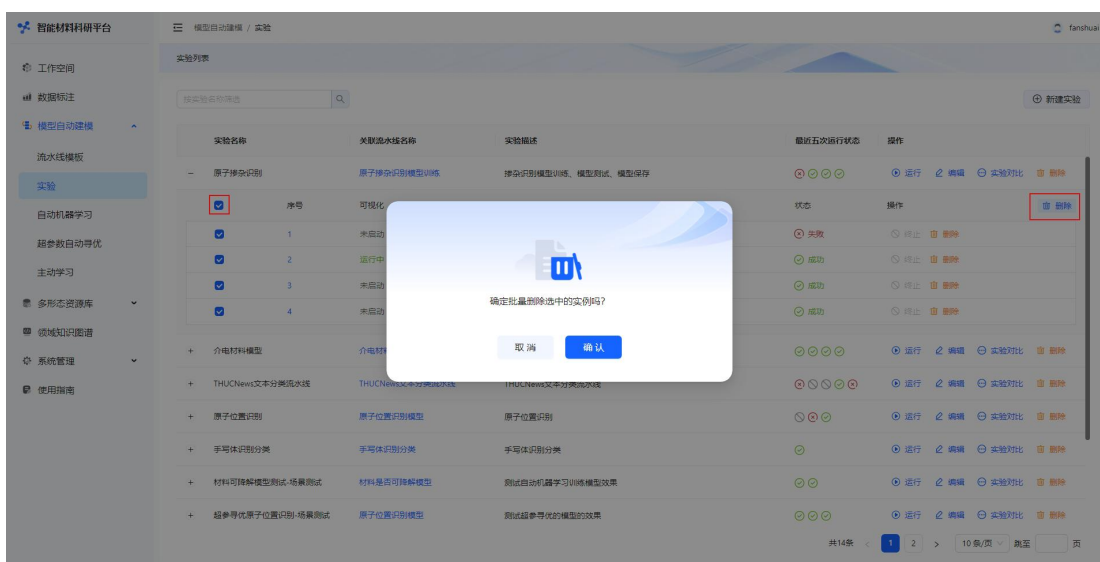


图 2-47 批量删除实验实例

## 2.6 自动机器学习

自动机器学习能够根据数据集和目标任务自动生成最优的模型方案, 大幅提高模型开发效率, 同时确保模型性能的优化, 使得非专家用户也能轻松构建高性能的机器学习模型。自动机器学习能够根据数据集和目标任务自动生成最优的模型方案, 大幅提高模型开发效率, 同时确保模型性能的优化, 使得非专家用户也能轻松构建高性能的机器学习模型。

### 2.6.1 查看自动机器学习模块

介绍用户如何进入自主机器学习实验列表界面, 可通过关键词快速检索目标

实验。

步骤:

1. 点击菜单栏【模型自动建模】-【自动机器学习】菜单进入自主机器学习实验列表界面，实验列表包括实验名称、实验描述、类型、更新时间、最近五次运行状态和操作，如图 2-48 所示。

2. 自主机器学习实验列表上方筛选框输入关键字，按 Enter 键将进行模糊搜索，列表只显示实验名称包括关键字的实验，如图 2-49 所示。

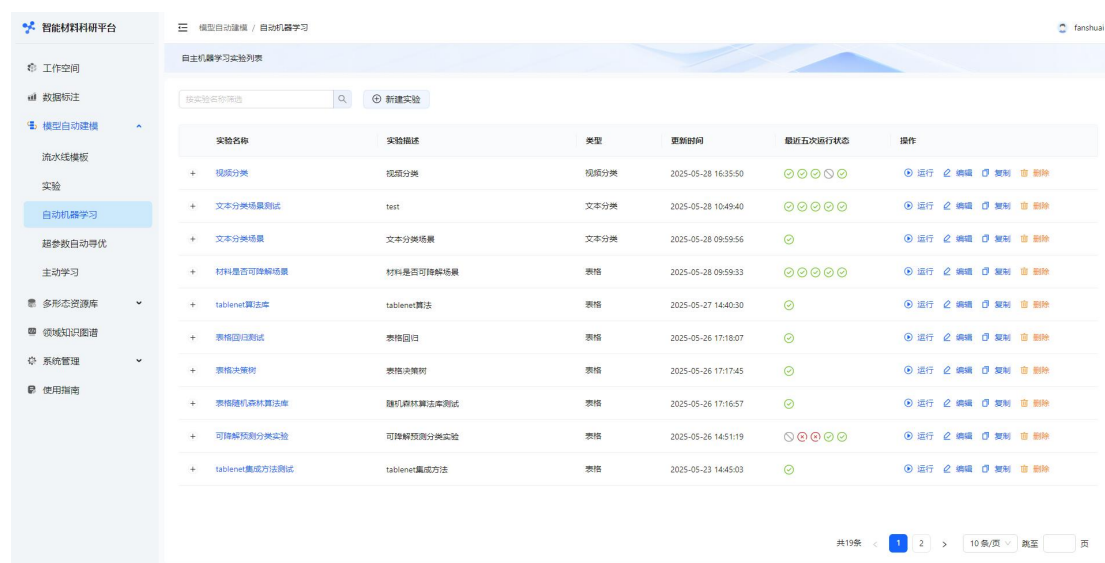


图 2-48 自动机器学习模块

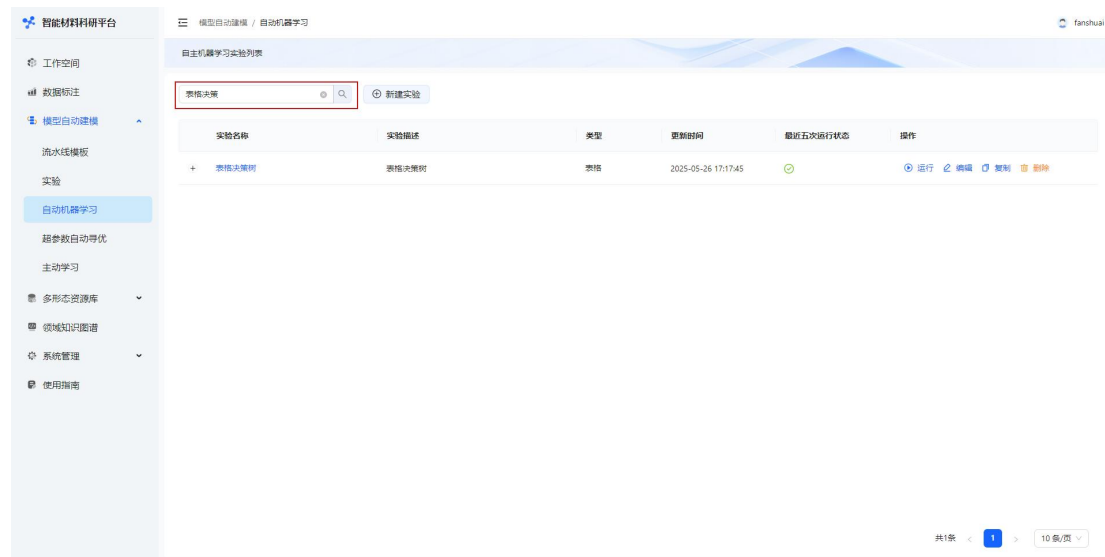


图 2-49 自动机器学习查询实验

## 2.6.2 新建实验

创建新的实验任务，提供对文本数据、表格数据、视频数据的自动化处理功

能。

### 步骤:

1. 点击【新建实验】按钮跳转到【新建实验】页面，如图 2-50 所示。
2. 输入基本信息，其中类型可选择【表格】、【文本分类】或【视频分类】，选择不同类型，基本信息下方显示相关配置。
3. 输入相关配置，点击页面【新建】按钮成功新建自动机器学习实验。

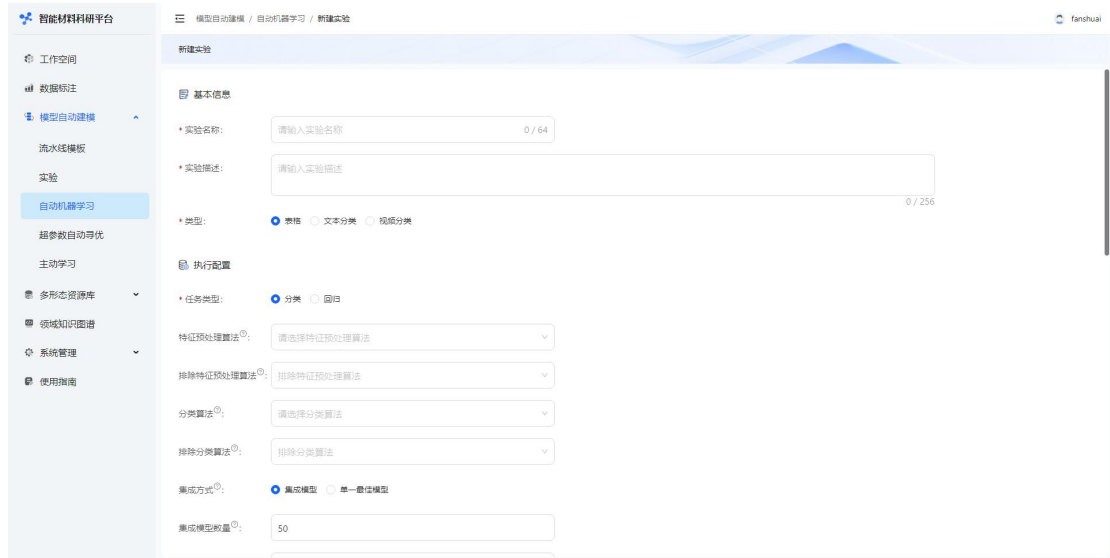


图 2-50 自动机器学习新建实验

### 2.6.3 编辑实验

对自动机器学习实验进行修改配置，包括修改预处理算法、集成方式、重采样策略、优化指标等。

### 步骤:

1. 自主机器学习实验列表点击【编辑】按钮跳转到【编辑实验】页面，如图 2-51 所示。
2. 编辑相关配置，点击页面【更新】按钮编辑实验保存成功。

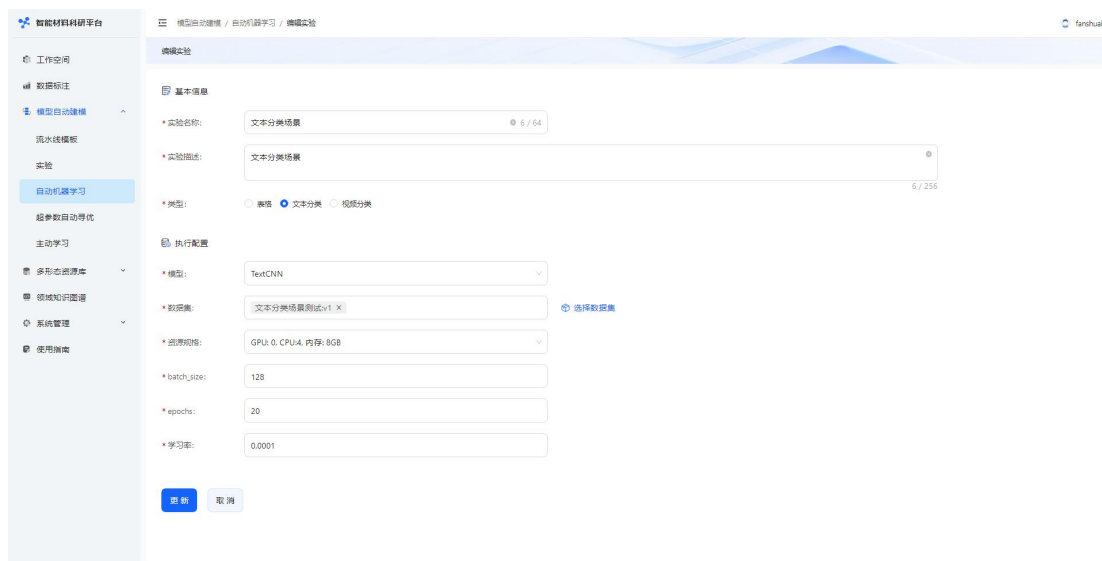


图 2-51 自动机器学习编辑实验

## 2.6.4 复制实验

通过复制现有自动机器学习实验快速创建新实验，保留原有配置，提高工作效率。

**步骤：**

1. 自主机器学习实验列表点击【复制】按钮跳转到【复制实验】页面，如图 2-52 所示。
2. 编辑相关配置，点击页面【确定】按钮复制实验保存成功。

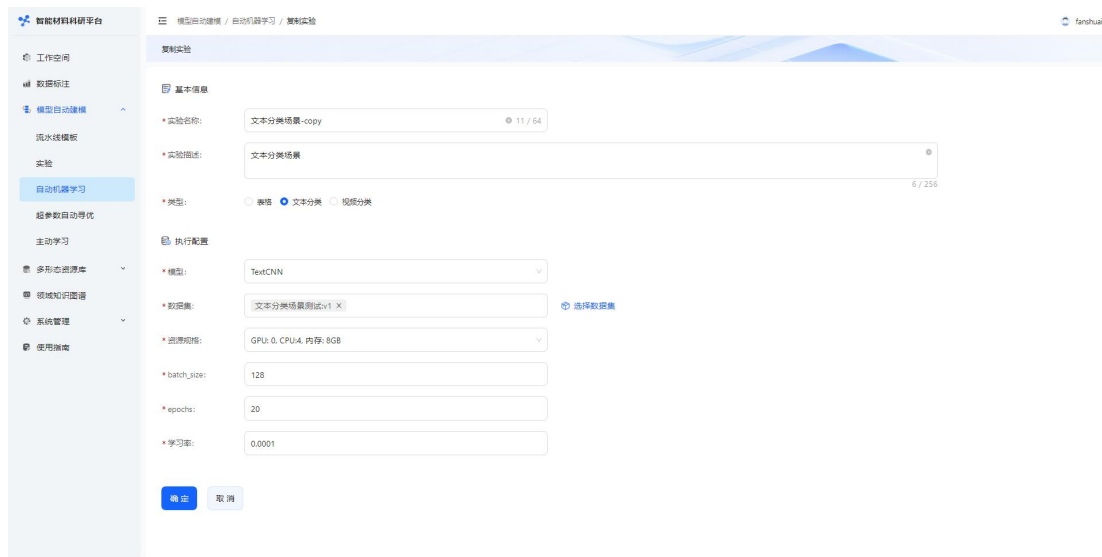


图 2-52 自动机器学习复制实验

## 2.6.5 运行实验

启动实验任务，生成实验实例，系统会自动分配资源并执行实验流程。

## 步骤:

自主机器学习实验列表点击【运行】按钮即可运行实验，展开实验实例列表自动生成一条实验实例，如图 2-53 所示。



图 2-53 自动机器学习运行实验

### 2.6.6 实验详情查看

查看实验运行信息、配置信息和日志等内容。

## 步骤:

1. 自主机器学习实验运行后，点击实验实例列表的【序号】，可以进入实验实例详情【基本信息】页面，基本信息包括运行信息和配置信息，如果实验运行成功则显示【实验结果】和【试验列表】（表格类型实验）/【可视化结果】（视频/文本分类类型实验），如图 2-54 所示。

2. 点击【日志】文本显示任务执行日志，如图 2-55 所示。

3. 点击【实验结果】显示自动机器学习实验结果和可视化结果，如图 2-56 所示。点击【模型下载】按钮可下载自动机器学习实验结果模型。

4. 如为【试验列表】点击显示对多次试验进行统计、排序，如图 2-57 所示。

如为【可视化结果】点击以图表可视化显示实验结果，如图 2-58 所示。



图 2-54 自动机器学习实验详情

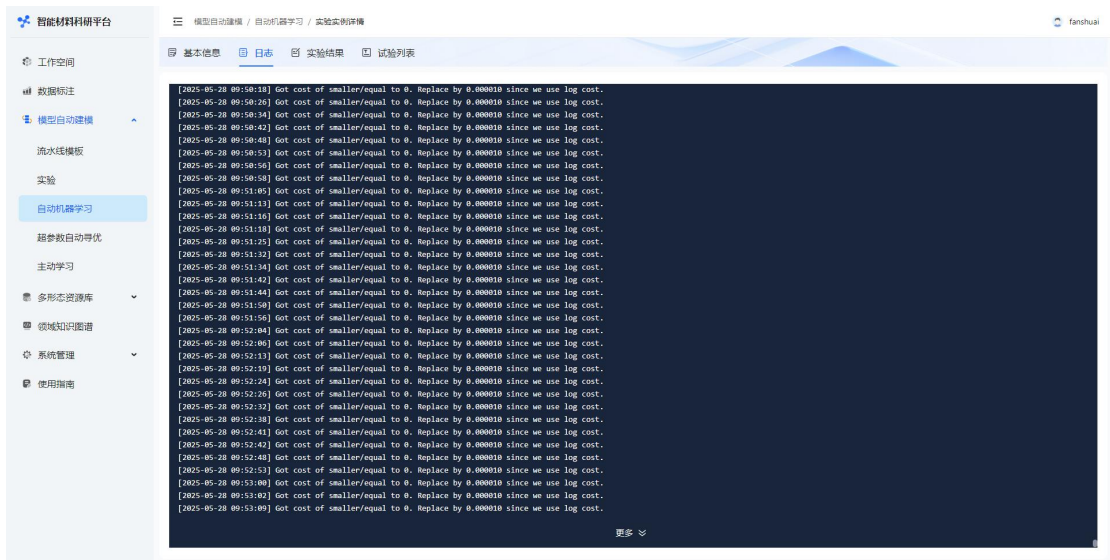


图 2-55 自动机器学习实验详情日志



图 2-56 自动机器学习实验详情实验结果

ID	耗时	训练损失	特征处理	算法	状态
1	3.108109474182129	0	no_preprocessing	random_forest	成功
2	3.630729913711548	0.0207939950850661602	fast_ica	mlp	成功
3	3.6415388584136963	0	select_percentile_classification	random_forest	成功
4	2.3256237506866455	0.047258979206049156	feature_agglomeration	mlp	成功
5	1.8633477687835693	0	feature_agglomeration	mlp	成功
6	--	--	extra_trees_preproc_for_classification	gradient_boosting	失败
7	3.003735422973633	0	extra_trees_preproc_for_classification	random_forest	成功
8	--	--	select_percentile_classification	gradient_boosting	失败
9	7.7573840618133545	0.001890359168241984	polynomial	extra_trees	成功
10	3.257136106491089	0.003700718336483968	extra_trees_preproc_for_classification	extra_trees	成功
11	--	--	extra_trees_preproc_for_classification	gradient_boosting	失败
12	1.9143319129943848	0.011342155009451793	select_rates_classification	extra_trees	成功
13	4.8940558433532715	0	feature_agglomeration	mlp	成功

图 2-57 自动机器学习实验详情试验列表

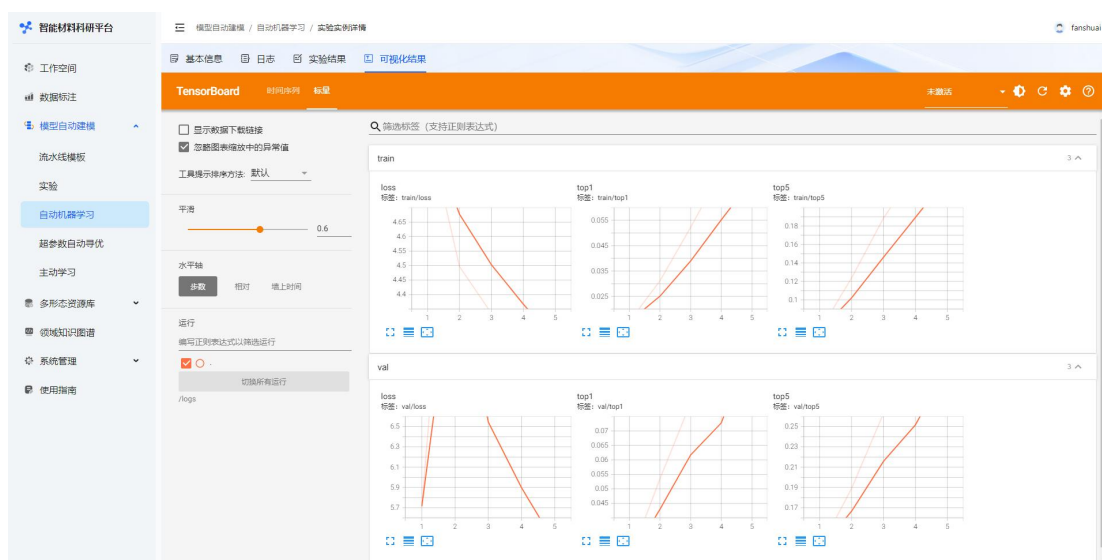


图 2-58 自动机器学习实验详情可视化结果

### 2.6.7 终止实验

手动终止正在运行的实验实例，释放计算资源，适用于实验异常或需要中断的场景。

#### 步骤：

实验实例列表点击【终止】-【确认】按钮，可成功终止正在运行的实验实例，如图 2-59 所示。

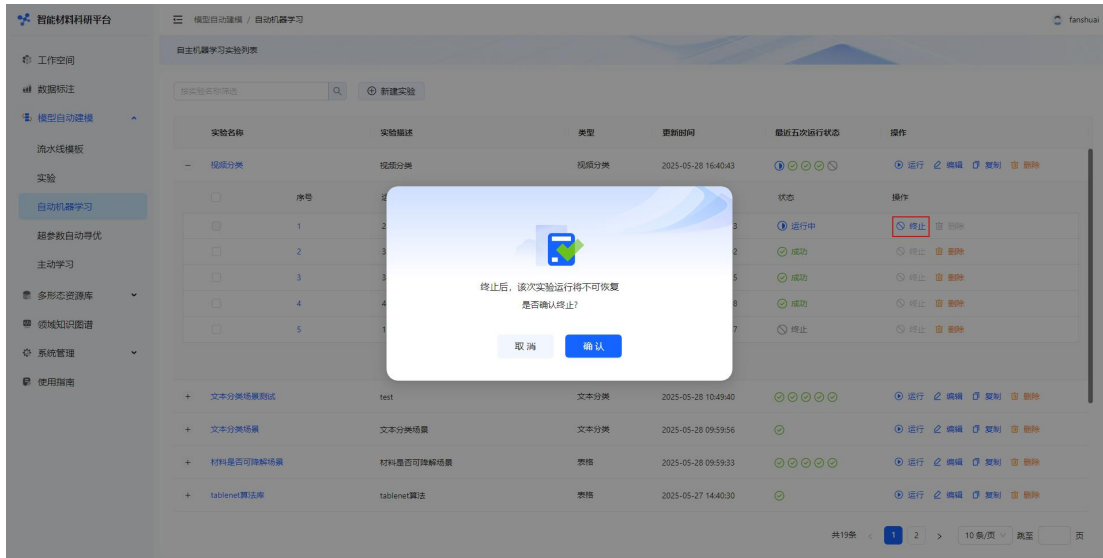


图 2-59 自动机器学习终止实验

### 2.6.8 删除实验

选择不再需要的实验进行删除操作，释放存储空间，同时系统会确保删除操作的安全性，当实验存在实验实例则无法删除，删除前进行提示，避免误删重要实验。

**步骤：**

1. 自主机器学习实验列表点击【删除】-【确认】按钮，如图 2-60 所示，如果实验无实验实例，则成功删除实验，如果存在实验实例则删除失败且弹框提示。
2. 点击【+】展开实验实例列表，勾选实验所有的实验实例，点击【删除】-【确认】按钮成功批量删除实验实例，如图 2-61 所示。实验列表再次点击【删除】-【确认】按钮，成功删除实验。

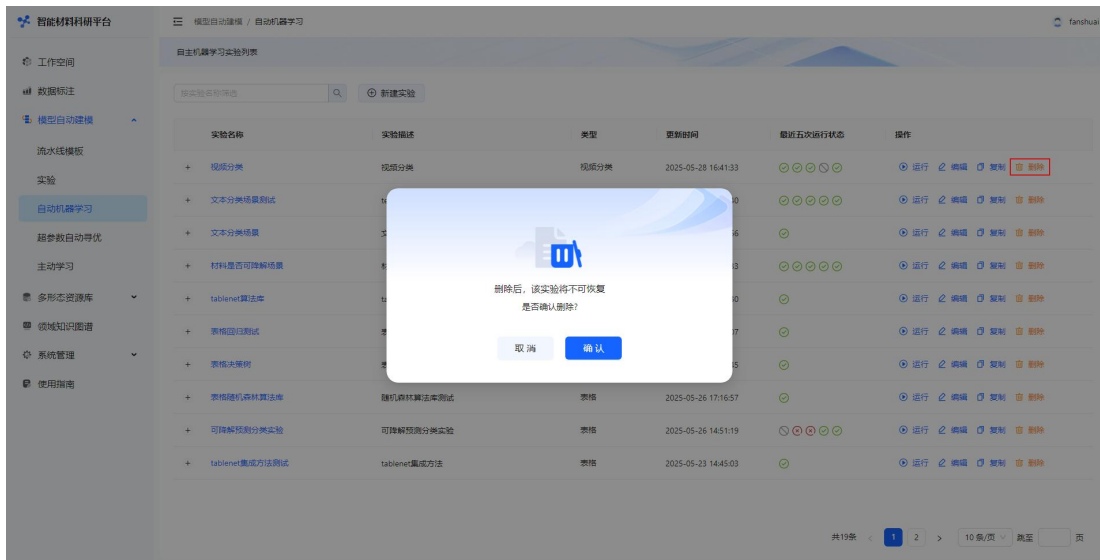


图 2-60 自动机器学习删除实验

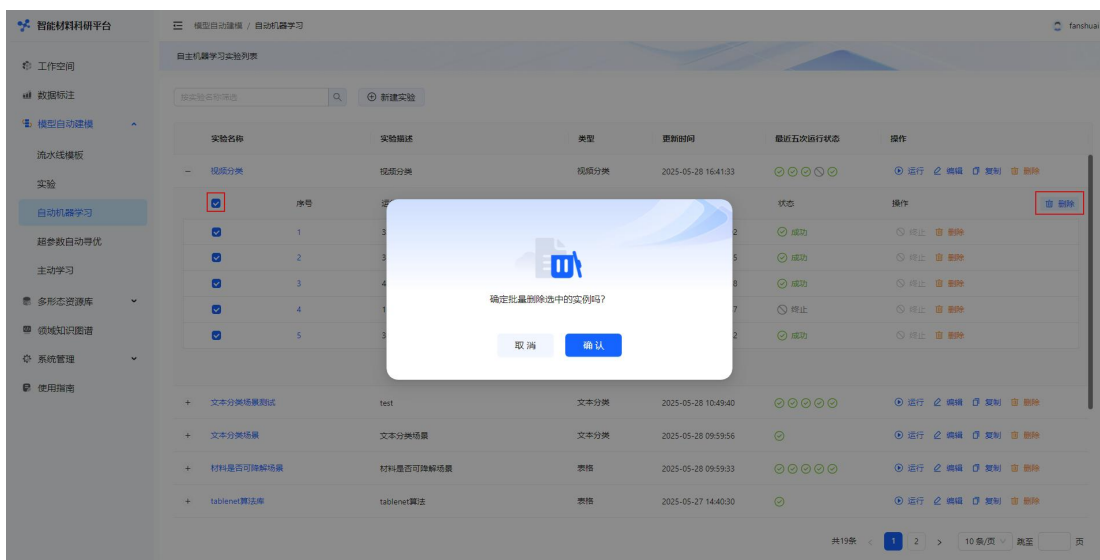


图 2-61 自动机器学习批量删除实验实例

## 2.7 超参数自动寻优

超参数自动寻优场景，支持对 PyTorch、TensorFlow、scikit-learn 等主流学习框架进行自动化参数调优。该模块内置多种优化算法，包括贝叶斯优化、网格搜索和种群优化，能够根据用户定义的超参数范围和优化目标（如准确率、F1 分数等），自动搜索最优参数组合并完成模型训练与性能评估。通过并行化计算和智能搜索策略，模块显著减少了手动调参的时间和工作量，适用于深度学习模型和传统机器学习模型的复杂超参数空间优化场景，帮助用户快速提升模型性能，降低开发成本。

## 2.7.1 查看超参数自动寻优模块

介绍用户如何进入超参数自动寻优实验列表界面，可通过关键词快速检索目标实验。

**步骤：**

1. 点击菜单栏【模型自动建模】-【超参数自动寻优】菜单进入超参数自动寻优实验列表界面，实验列表包括实验名称、实验描述、更新时间、最近五次运行状态和操作，如图 2-62 所示。

2. 超参数自动寻优实验列表上方筛选框输入关键字，按 Enter 键将进行模糊搜索，列表只显示实验名称包括关键字的实验，如图 2-63 所示。

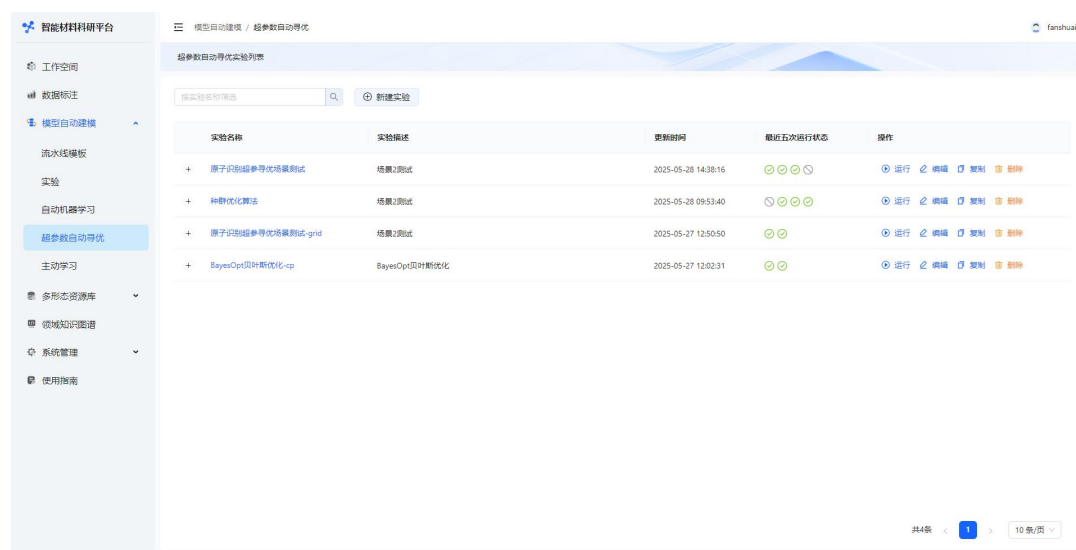


图 2-62 超参数自动寻优模块



图 2-63 超参数自动寻优查询实验

## 2.7.2 新建实验

创建新的超参数自动寻优实验任务。

**步骤：**

1. 点击【新建实验】按钮跳转到【新建实验】页面，如图 2-64 所示。
2. 输入实验的基本信息，并配置实验的具体参数。
3. 点击页面【新建】按钮成功新建超参数自动寻优实验。

图 2-64 超参数自动寻优新建实验

## 2.7.3 编辑实验

调整实验的配置，修改完成后系统保存更新后的实验配置，用户可以重新运行实验以应用更改。

**步骤：**

1. 超参数自动寻优实验列表点击【编辑】按钮跳转到【编辑实验】页面，如图 2-65 所示。
2. 编辑相关配置，点击页面【更新】按钮编辑实验保存成功。

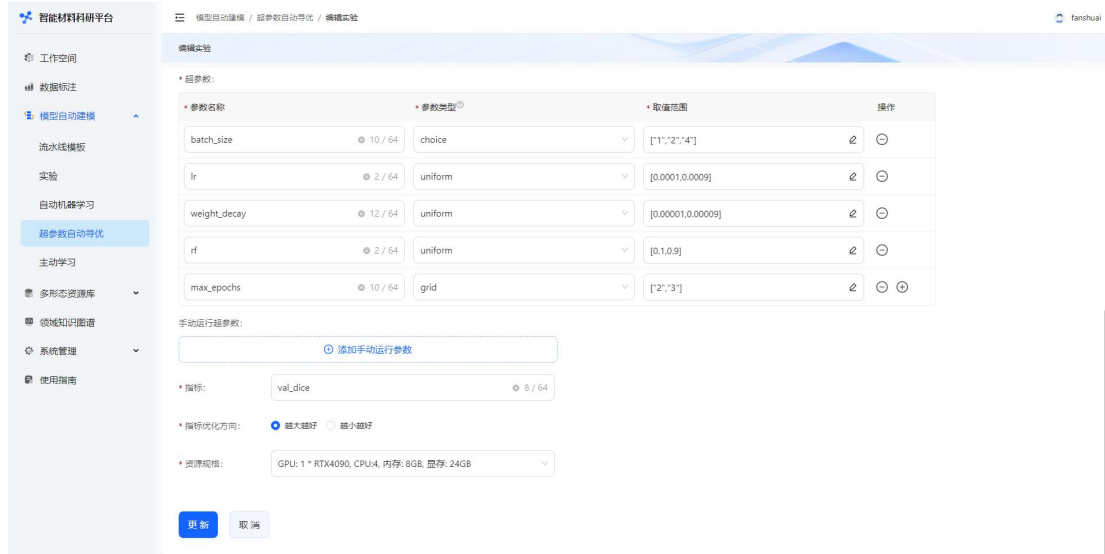


图 2-65 超参数自动寻优编辑实验

### 2.7.4 复制实验

通过复制现有超参数自动寻优实验快速创建新实验，保留原有配置，提高工作效率。

**步骤：**

1. 超参数自动寻优实验列表点击【复制】按钮跳转到【复制实验】页面，如图 2-66 所示。
2. 编辑相关配置，点击页面【确定】按钮复制实验保存成功。



图 2-66 超参数自动寻优复制实验

## 2.7.5 运行实验

启动实验任务，生成实验实例，系统根据实验配置自动分配资源并启动实验实例，平台实时监控实验状态和性能指标，同时根据用户定义的超参数范围和优化目标，自动搜索最优参数组合并完成模型训练与性能评估。

### 步骤：

超参数自动寻优实验列表点击【运行】按钮即可运行实验，展开实验实例列表自动生成一条实验实例，如图 2-67 所示。

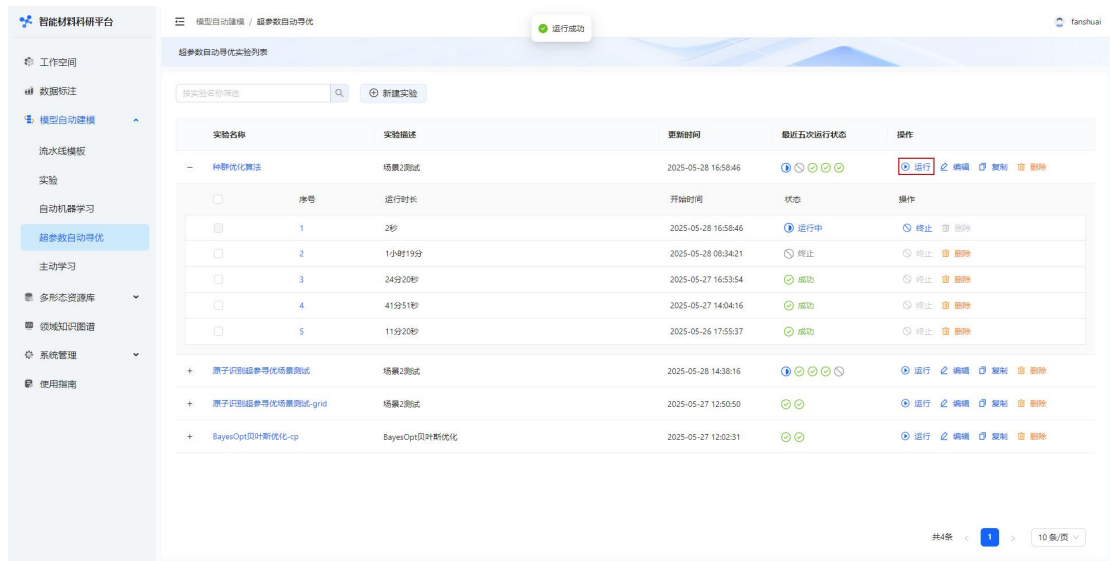


图 2-67 超参数自动寻优运行实验

## 2.7.6 实验详情查看

查看实验实例的详细运行信息，包括配置信息（如代码配置、数据集、模型、搜索算法等）、运行日志（如错误信息和调试日志）、性能指标（如训练损失、验证准确率等）以及资源使用情况。

### 步骤：

1. 超参数自动寻优实验运行后，点击实验实例列表的【序号】，可以进入实验实例详情【基本信息】页面，基本信息包括运行信息和配置信息，如果实验运行成功则显示【实验结果】和【寻优列表】，如图 2-68 所示。

2. 点击【日志】文本显示任务执行日志，包括系统日志和超参寻优日志，如图 2-69 所示。

3. 点击【实验结果】显示最佳实验结果，如图 2-70 所示。

4. 点击【寻优列表】显示超参数寻优列表，如图 2-71 所示，列表包括最佳

寻优结果，并可进行寻优结果下载。

**运行信息**

启动时间: 2025-05-27 12:39:33      执行时长: 11分15秒      状态: 成功

**配置信息**

代码配置: 原子识别代码      主函数代码文件: train\_rl.py      镜像: 172.20.32.187/machine-learning/xy\_atom\_recogni...  
 数据集: 原子位置识别训练数据集-拓普测试v1      模型: --      总试验次数: 2  
 搜索算法: --      调度算法: --      单次试验最大时间: --  
 最小试验数: --      指标优化方向: 越大越好      指标: val\_dice  
 资源规格: GPU: 1 \* RTX4090, CPU: 4, 内存: 8GB, 显存: 24GB

超参数

参数名称	参数类型	取值范围
batch_size	choice	['1','2','4']
lr	uniform	[0.0001,0.0009]
weight_decay	uniform	[0.00001,0.00009]
rf	uniform	[0,1,0.9]
max_epochs	grid	['2','3']

图 2-68 超参数自动寻优实验详情

系统日志    超参数寻优日志

```
[2025-05-27 12:40:00] time="2025-05-27T12:40:00.848Z" level=info msg="capturing logs" argo=true
[2025-05-27 12:40:00] Cloning into '/tmp/traincode'...
[2025-05-27 12:41:18] Updating files: 51% (10/31)Updating files: 58% (18/31)Updating files: 64% (20/31)Updating files: 67% (21/31)Updating files:
70% (22/31)Updating files: 74% (23/31)Updating files: 77% (24/31)Updating files: 80% (25/31)Updating files: 83% (26/31)Updating files: 87% (27/31)Updating files: 90% (28/31)Updating files: 93% (29/31)Updating files: 96% (30/31)Updating files: 98% (31/31)Updating files: 100% (31/31), done.
[2025-05-27 12:41:18] Cloning public repository without authentication.
[2025-05-27 12:41:18] Repository cloned successfully.
[2025-05-27 12:41:18] time="2025-05-27T12:41:18.888Z" level=info msg="sub-process exited" argo=true error="cnlls"
```

更多

图 2-69 超参数自动寻优实验详情日志



图 2-70 超参数自动寻优实验详情最佳实验结果

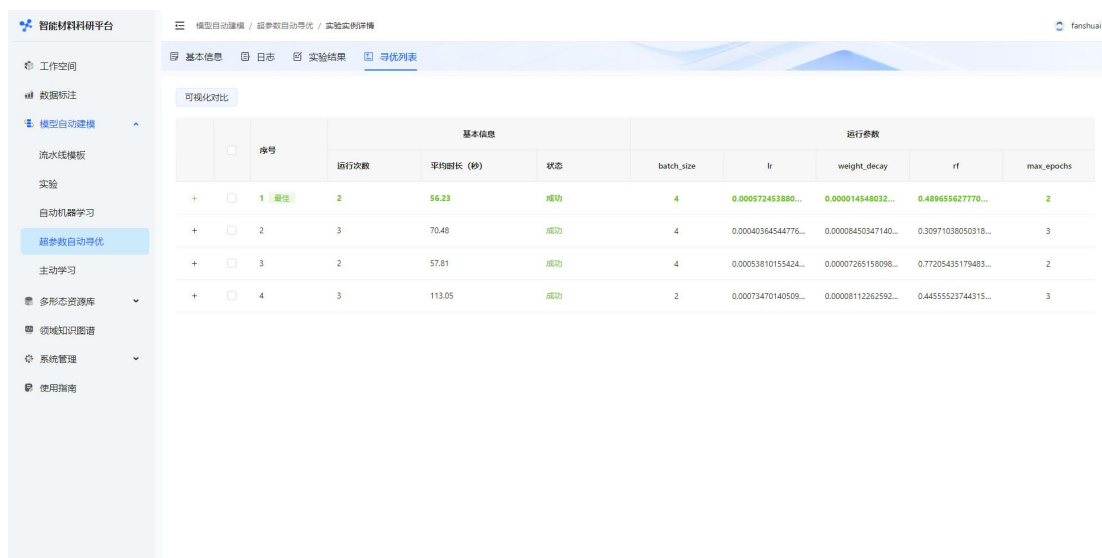


图 2-71 超参数自动寻优实验详情寻优列表

### 2.7.7 终止实验

手动终止正在运行的实验实例，释放计算资源，适用于实验异常或需要中断的场景。

#### 步骤：

实验实例列表点击【终止】-【确认】按钮，可成功终止正在运行的实验实例，如图 2-72 所示。

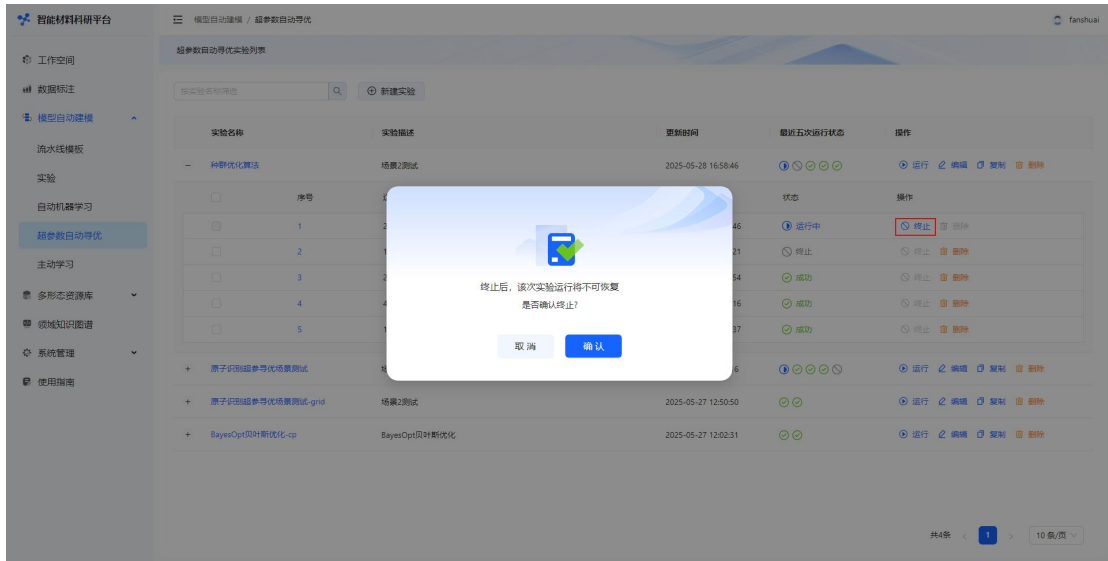


图 2-72 超参数自动寻优终止实验

### 2.7.8 删除实验

选择不再需要的实验进行删除操作，释放存储空间，同时系统会确保删除操作的安全性，当实验存在实验实例则无法删除，删除前进行提示，避免误删重要实验。

#### 步骤：

1. 超参数自动寻优实验列表点击【删除】-【确认】按钮，如图 2-73 所示，如果实验无实验实例，则成功删除实验，如果存在实验实例则删除失败且弹框提示。
2. 点击【+】展开实验实例列表，勾选实验所有的实验实例，点击【删除】-【确认】按钮成功批量删除实验实例，如图 2-74 所示。实验列表再次点击【删除】-【确认】按钮，成功删除实验。

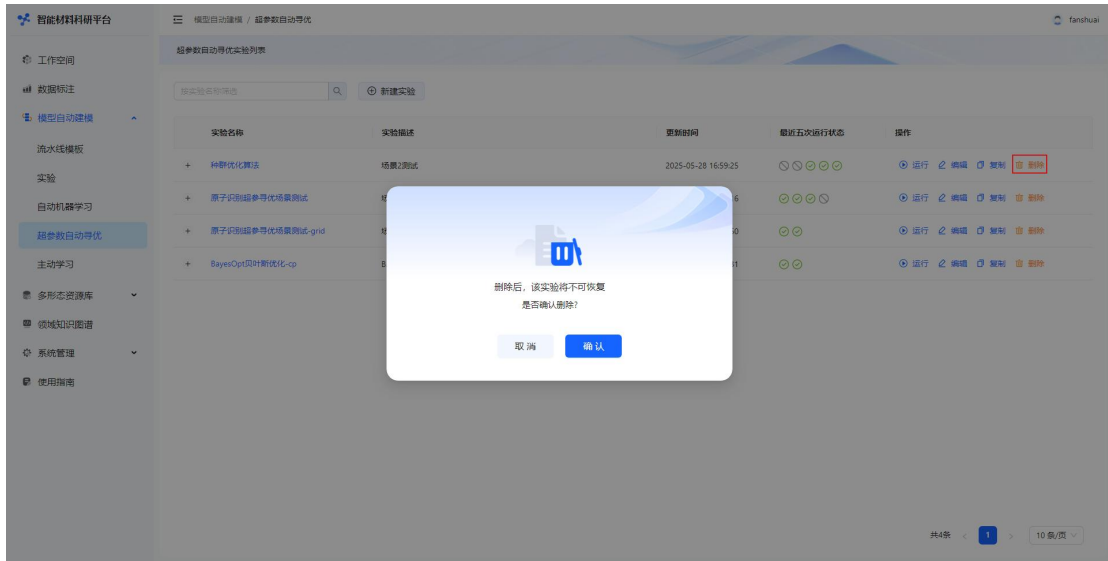


图 2-73 超参数自动寻优删除实验

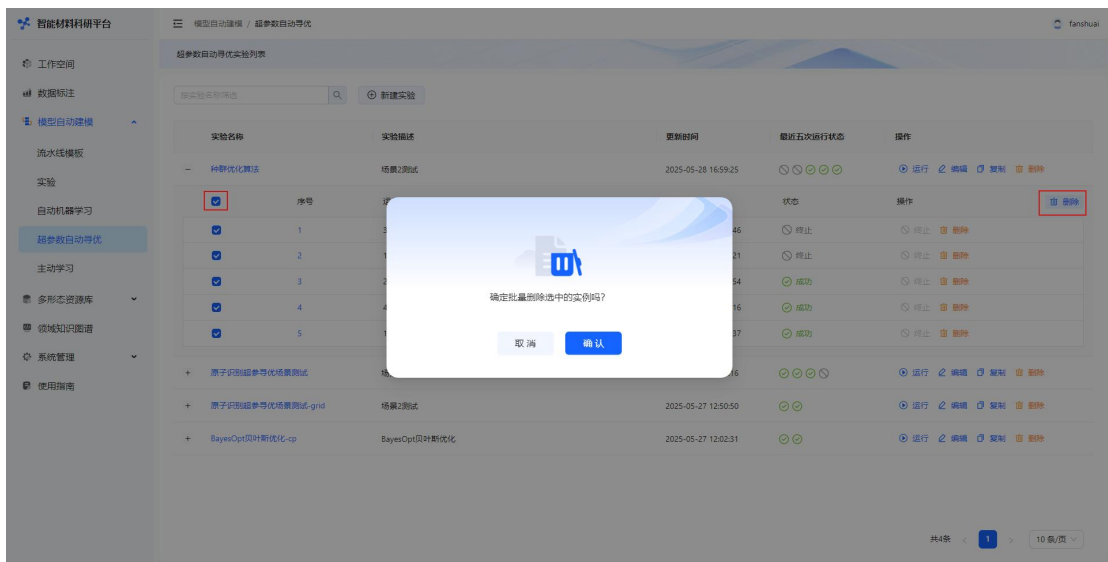


图 2-74 超参数自动寻优批量删除实验实例

## 2.8 主动学习

主动学习通过智能筛选对模型最有价值的样本（如分类边界模糊或预测置信度低的样本），交由人工标注，从而减少标注成本并提升模型性能。

### 2.8.1 查看主动学习模块

介绍用户如何进入自动学习实验列表界面，可通过关键词快速检索目标实验。

**步骤：**

1. 点击菜单栏【模型自动建模】-【主动学习】菜单进入自动学习实验列表界面，实验列表包括实验名称、实验描述、更新时间、最近五次运行状态和操作，

如图 2-75 所示。

2. 主动学习实验列表上方筛选框输入关键字，按 Enter 键将进行模糊搜索，列表只显示实验名称包括关键字的实验，如图 2-76 所示。

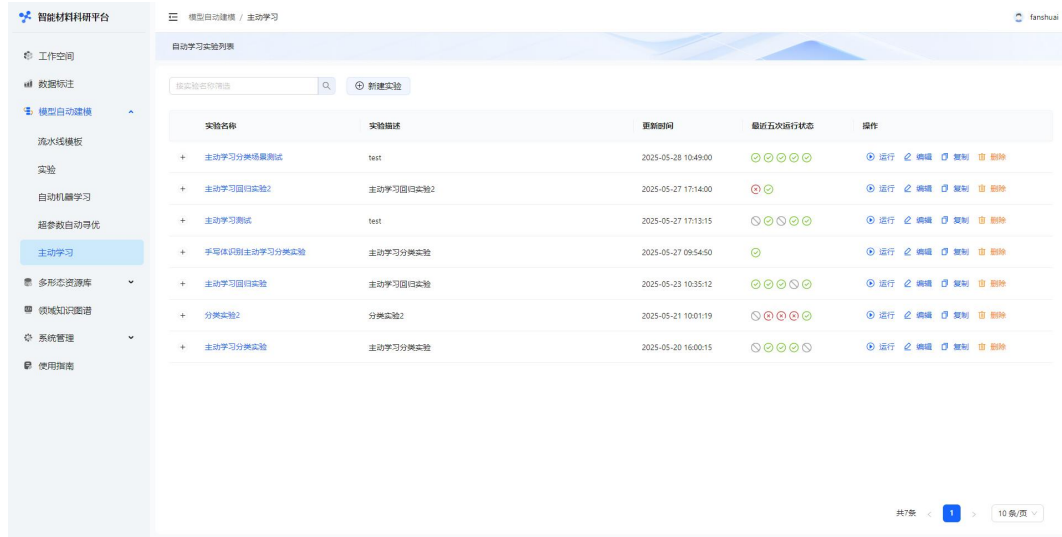


图 2-75 主动学习模块

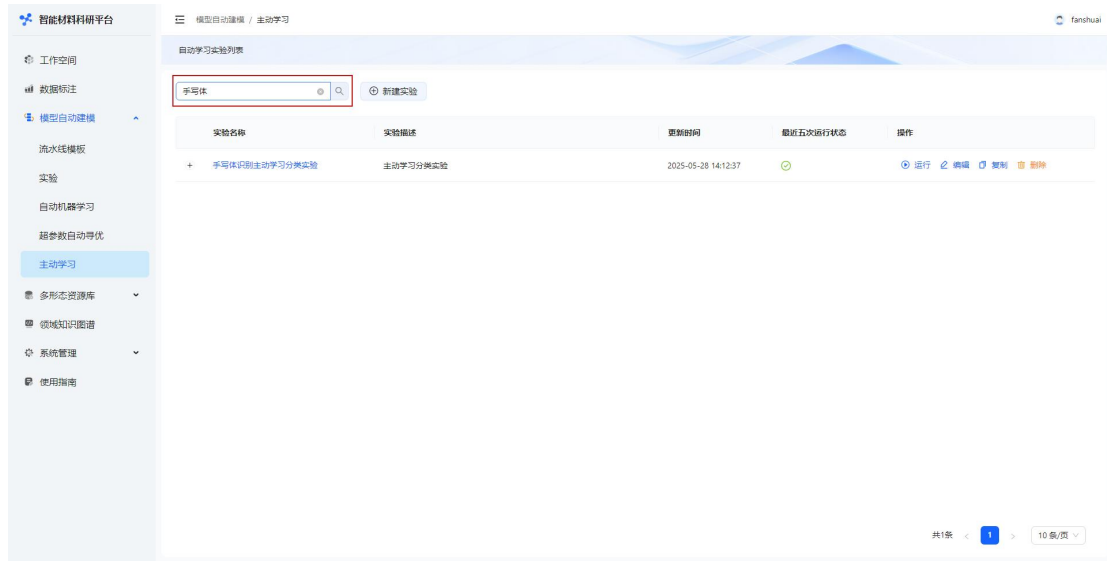


图 2-76 主动学习查询实验

### 2.8.2 新建实验

创建新的自动学习实验任务。

**步骤：**

1. 点击【新建实验】按钮跳转到【新建实验】页面，如图 2-77 所示。
2. 输入实验的基本信息和配置信息。
3. 点击页面【新建】按钮成功新建自动学习实验。

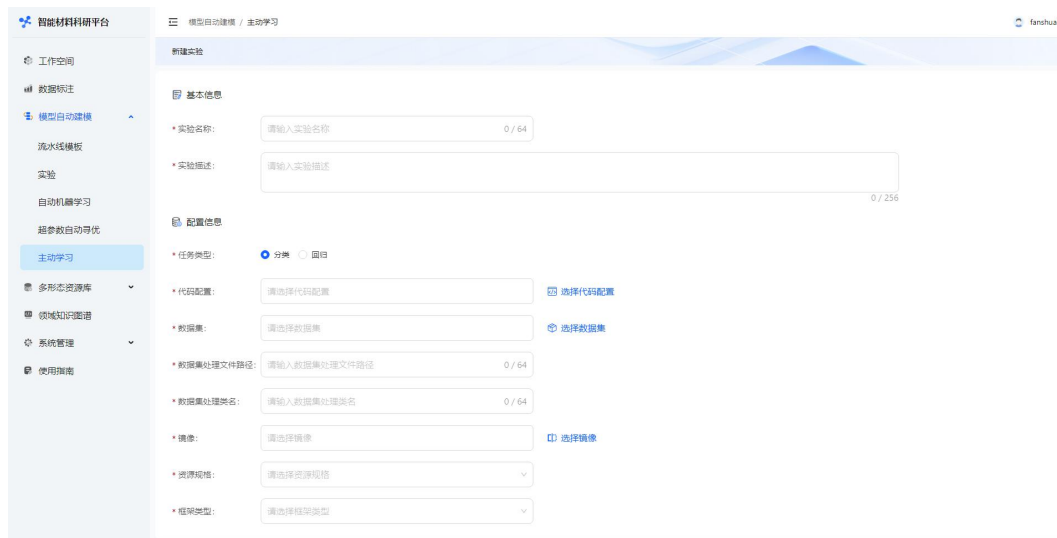


图 2-77 主动学习新建实验

### 2.8.3 编辑实验

对已创建的主动学习任务进行参数修改,修改后用户可以重新运行实验以应用更改。

**步骤:**

1. 自动学习实验列表点击【编辑】按钮跳转到【编辑实验】页面,如图 2-78 所示。
2. 编辑相关配置,点击页面【更新】按钮编辑实验保存成功。

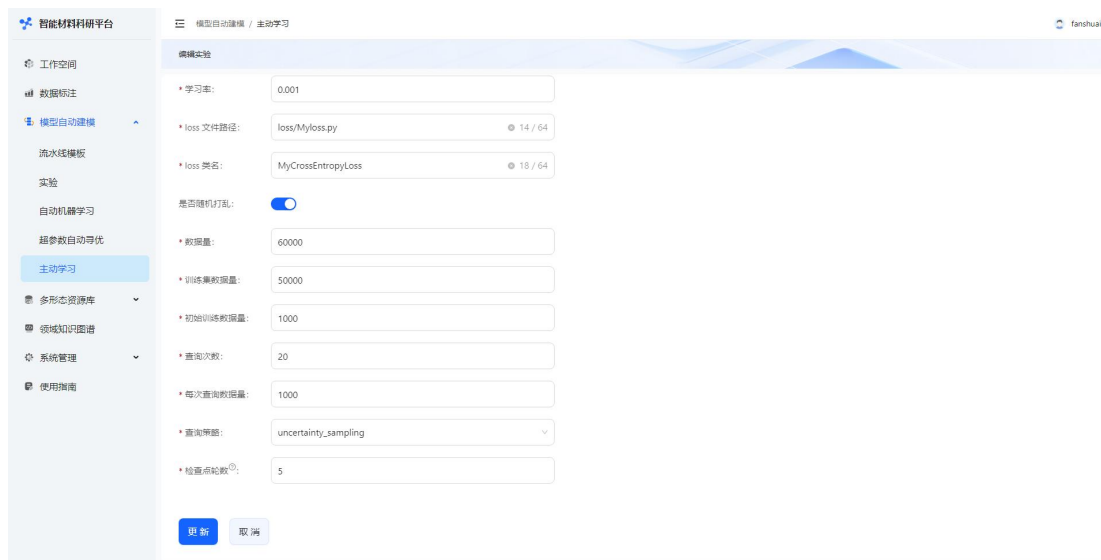


图 2-78 主动学习编辑实验

### 2.8.4 复制实验

通过复制现有主动学习实验快速创建新实验,保留原有配置,提高工作效率。

**步骤:**

1. 自动学习实验列表点击【复制】按钮跳转到【复制实验】页面，如图 2-79 所示。

2. 编辑相关配置，点击页面【确定】按钮复制实验保存成功。

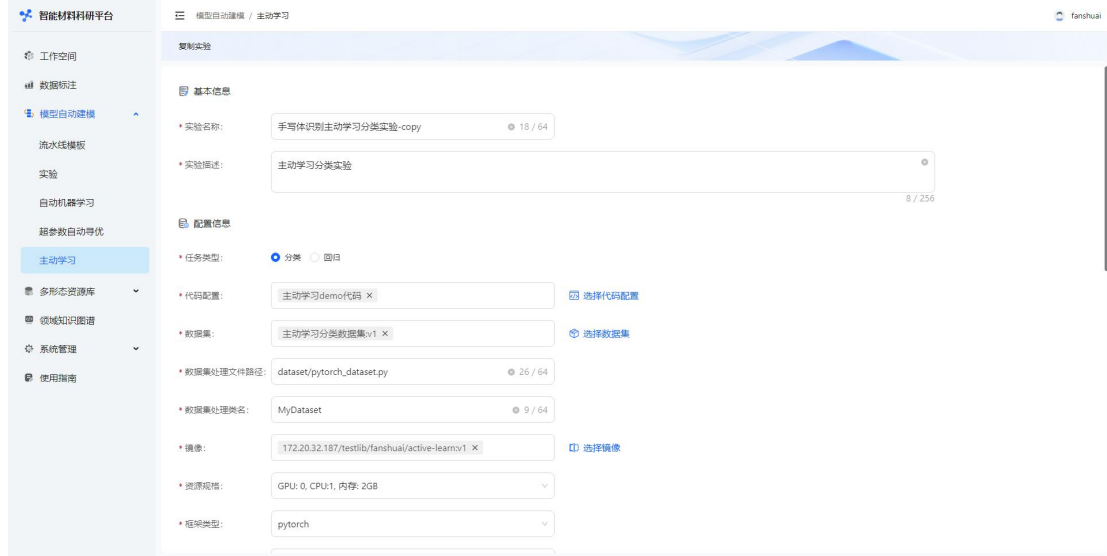


图 2-79 主动学习复制实验

### 2.8.5 运行实验

启动已创建的自动学习实验，实例运行后，系统根据配置筛选样本并生成训练任务。

**步骤：**

自动学习实验列表点击【运行】按钮即可运行实验，展开实验实例列表自动生成一条实验实例，如图 2-80 所示。

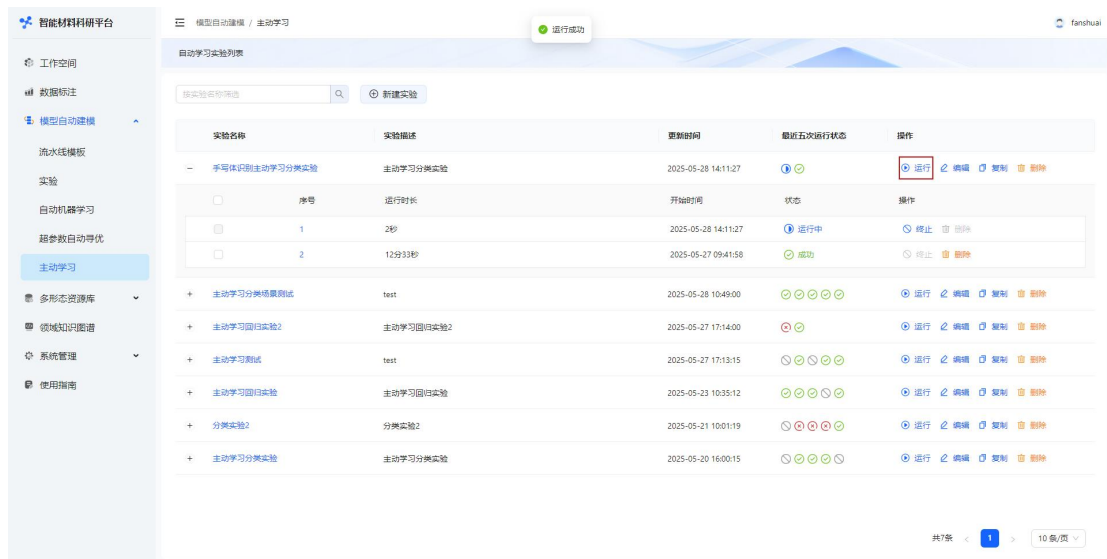


图 2-80 主动学习运行实验

## 2.8.6 实验详情查看

用户查看主动学习实验实例的详细信息，包括实例配置参数、筛选样本列表、已标注数据、模型训练历史记录、当前模型性能指标等。

**步骤：**

1. 主动学习实验运行后，点击实验实例列表的【序号】，可以进入实验实例详情【基本信息】页面，基本信息包括运行信息和配置信息，如果实验运行成功则显示【实验结果】、【训练列表】和【可视化结果】，如图 2-81 所示。

2. 点击【日志】文本显示任务执行日志，包括系统日志和主动学习日志，如图 2-69 所示。

3. 点击【实验结果】可数据选择下载或模型下载，如图 2-83 所示。

4. 点击【训练列表】显示数据训练列表，如图 2-84 所示。

5. 点击【可视化结果】对训练结果图表可视化展示，如图 2-85 所示。

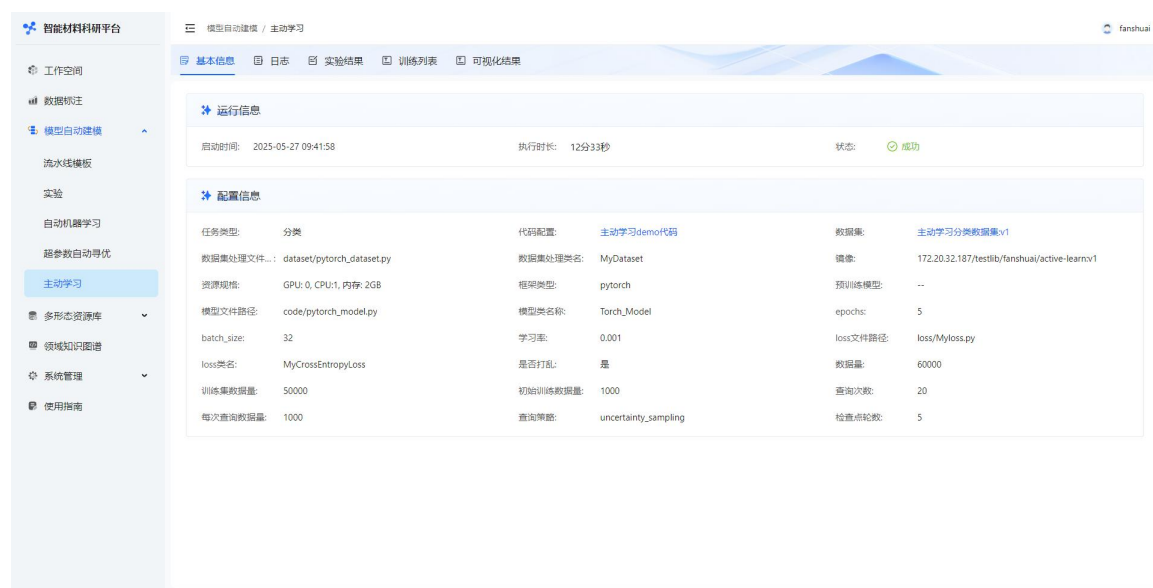


图 2-81 主动学习实验详情

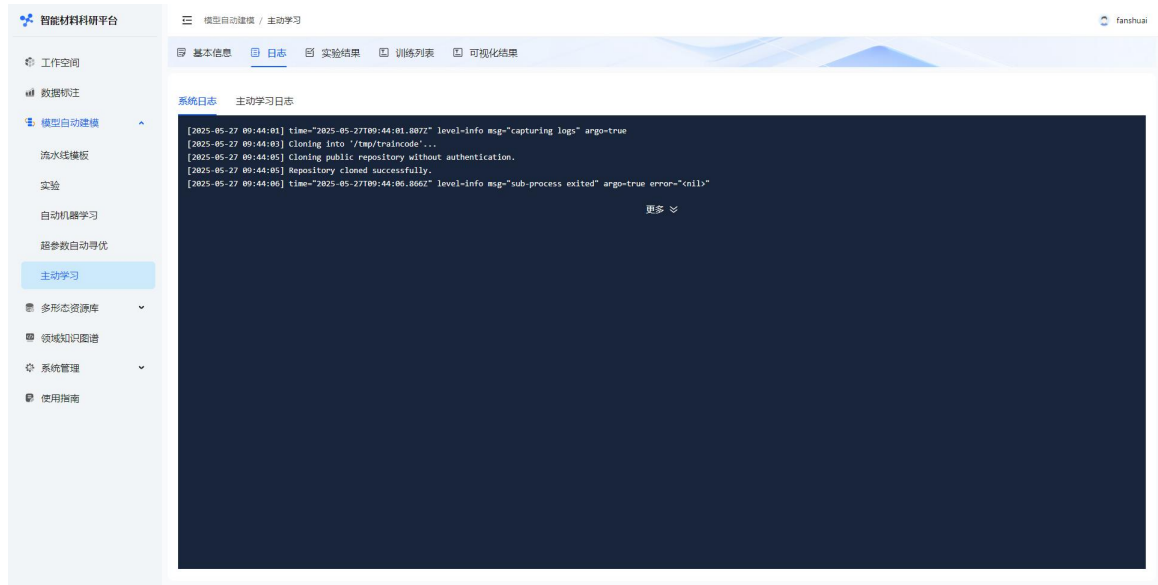


图 2-82 主动学习实验详情日志

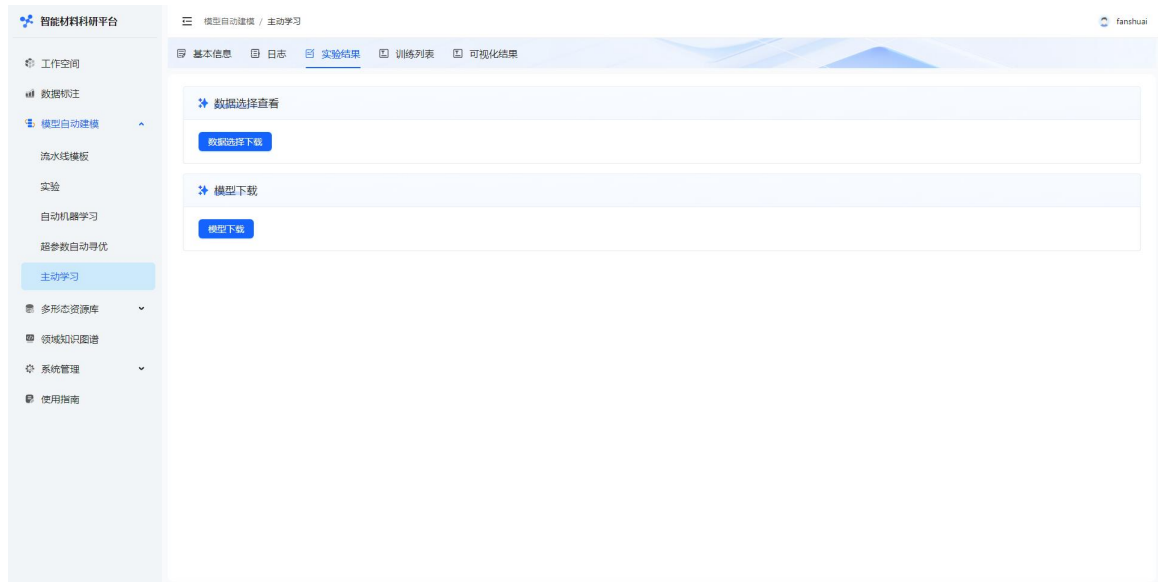


图 2-83 主动学习实验详情实验结果

序号	查询数取值	总数取值	准确率	查询数据 id
1	1000	2000	0.9497	下载
2	1000	3000	0.9698	下载
3	1000	4000	0.9799	下载
4	1000	5000	0.9798	下载
+	5	6000	0.9845	下载
6	1000	7000	0.9882	下载
7	1000	8000	0.9876	下载
8	1000	9000	0.9887	下载
9	1000	10000	0.989	下载
+	10	11000	0.9892	下载
11	1000	12000	0.9893	下载
12	1000	13000	0.9901	下载
13	1000	14000	0.9902	下载

图 2-84 主动学习实验详情训练列表



图 2-85 主动学习实验详情可视化结果

### 2.8.7 终止实验

支持用户终止正在运行的主动学习实例，终止后，实例停止筛选样本和生成训练任务。

#### 步骤：

实验实例列表点击【终止】-【确认】按钮，可成功终止正在运行的实验实例，如图 2-72 所示。

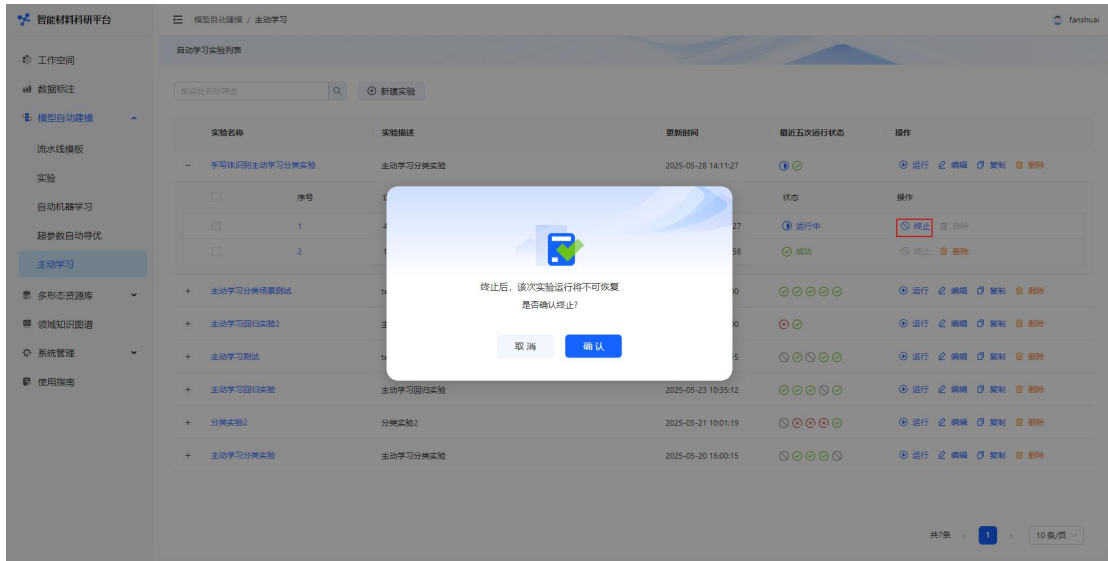


图 2-86 主动学习终止实验

### 2.8.8 删除实验

删除已创建的主动学习任务，删除后，任务相关的数据、模型状态和实验记录将被清除。

**步骤：**

1. 主动学习实验列表点击【删除】-【确认】按钮，如图 2-87 所示，如果实验无实验实例，则成功删除实验，如果存在实验实例则删除失败且弹框提示。

2. 点击【+】展开实验实例列表，勾选实验所有的实验实例，点击【删除】-【确认】按钮成功批量删除实验实例，如图 2-88 所示。

实验列表再次点击【删除】-【确认】按钮，成功删除实验。

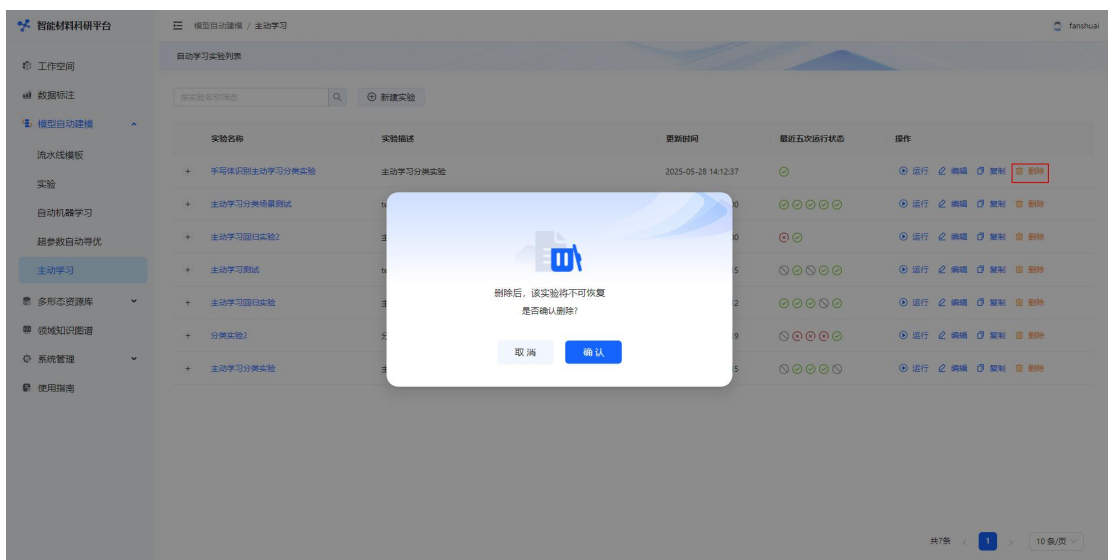


图 2-87 主动学习删除实验

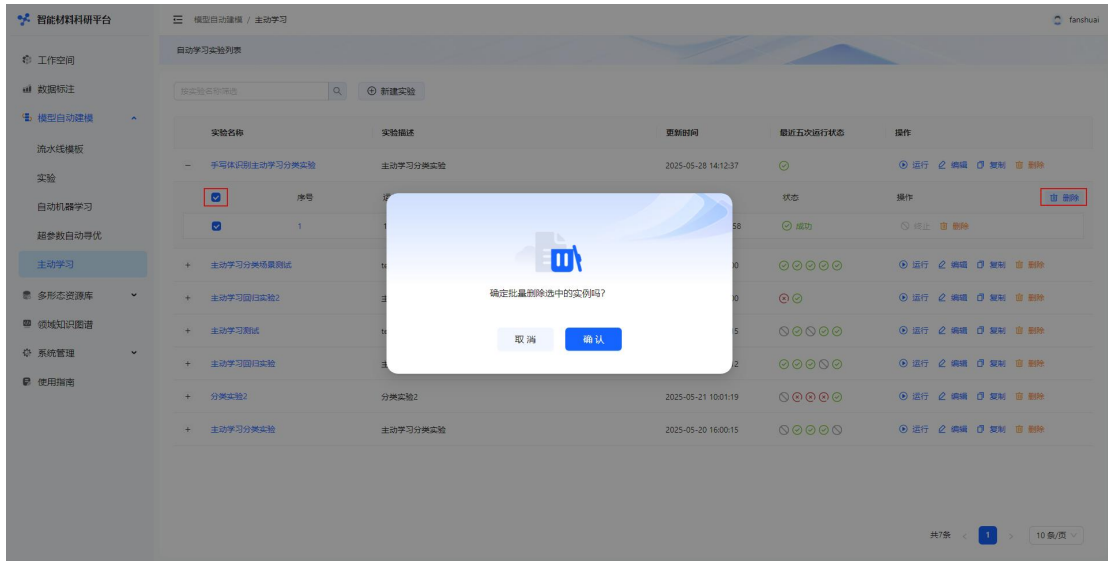


图 2-88 主动学习批量删除实验实例

## 2.9 数据集管理

数据集管理包含数据集广场和个人数据集。数据集广场汇聚了深度学习的常用的数据集。个人数据集则是用户上传的数据集。此功能是实验功能的基础，实验功能依赖于数据集管理功能。用户选择了对应的数据集，即可在此数据集上进行相应的流水线自动实验，以及实验后导出新的结果数据集。

### 2.9.1 查看数据集管理模块

介绍用户如何进入数据集管理模块界面，分为公开数据集（数据广场）和个人私有数据集，数据集卡片根据点赞数排序推荐，可通过关键词、标签等条件快速检索目标数据集。

**步骤：**

1. 1.点击菜单栏【多形态资源库】-【数据集】菜单进入数据广场界面，如图 2-89 所示。
2. 点击侧边栏【分类】或【研究方向/应用领域】标签，界面显示对应标签的数据集。
3. 数据集名称筛选框输入关键字，按 Enter 键或点击搜索图标将进行模糊搜索，界面只显示数据集名称包括关键字的数据集，如图 2-90 所示。



图 2-89 数据集管理模块



图 2-90 查询数据集

## 2.9.2 新建数据集

将新数据集添加到平台，并为其添加描述、标签等信息，便于后续管理和使用。

**步骤：**

1. 点击【个人数据】-【新建数据集】按钮弹出【新建数据集】弹窗，如图 2-91 所示。

2. 输入数据集名称、数据集版本、数据集描述、版本描述，选择可见性和上传数据集文件，选填数据集分类和研究方向/应用领域，点击弹窗【确定】按钮成功新建数据集。

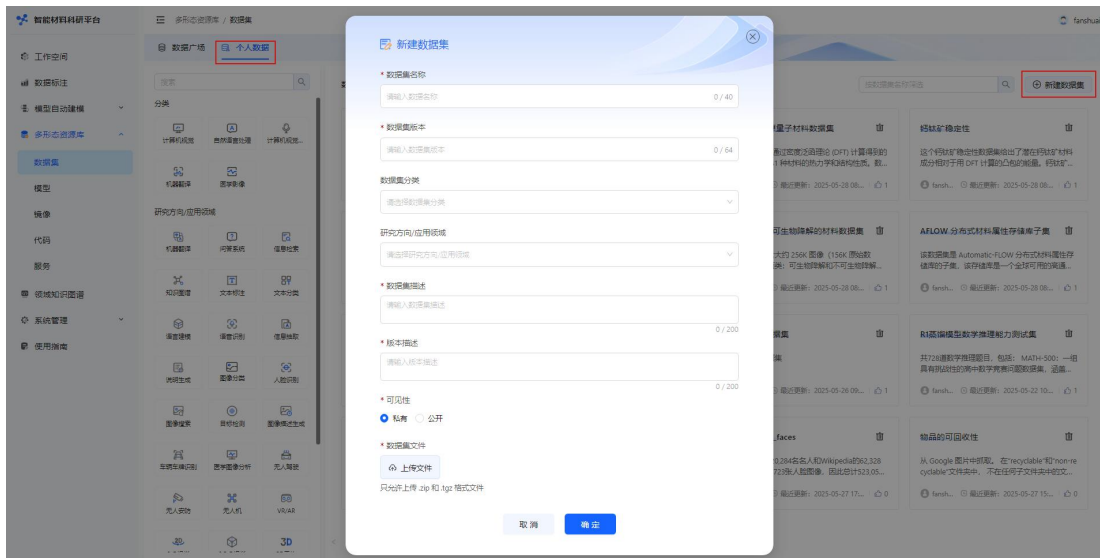


图 2-91 新建数据集

### 2.9.3 查看数据集

选择数据集版本后，查看数据集的详细版本信息，包括版本号、更新时间等元数据。

步骤：

1. 点击【数据集卡片】跳转到数据集简介界面，用户可查看数据集的名称，版本，创建人，更新时间等基本信息，如图 2-92 所示。
2. 点击【数据集文件】显示数据集版本的文件和文件大小等内容，如图 2-93 所示。
3. 点击【版本号】下拉框可选择切换数据集不同版本查看。



图 2-92 数据集简介

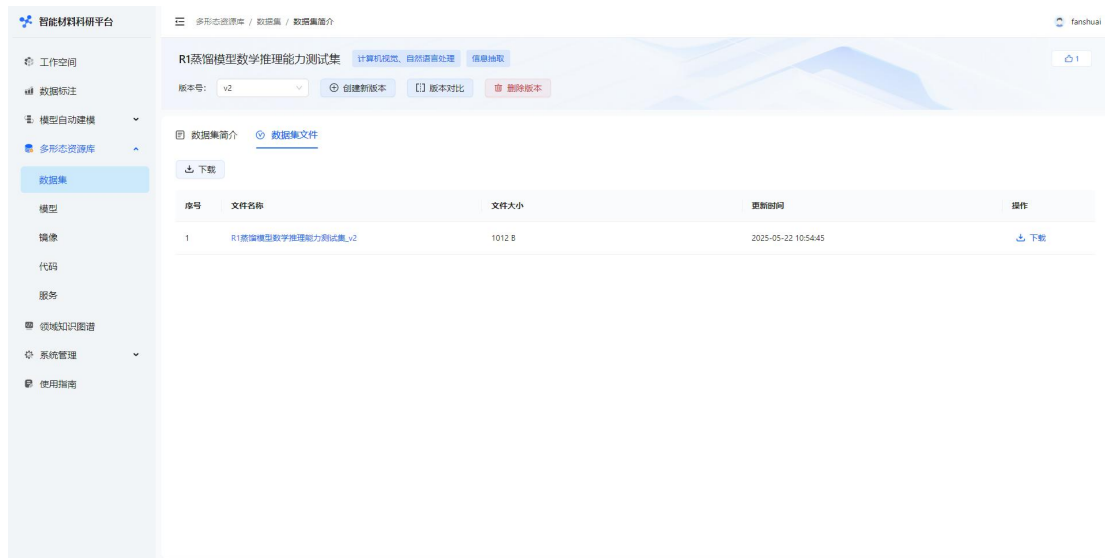


图 2-93 数据集文件

## 2.9.4 创建新版本

创建数据集新版本实现数据变更的精准管控与实验可复现性。

**步骤：**

1. 点击【数据集卡片】跳转到数据集简介界面。
2. 点击【创建新版本】按钮弹出创建新版本弹窗，如图 2-94 所示。
3. 输入数据集版本和版本描述，上传数据集文件，点击弹窗【确定】按钮成功创建新版本。

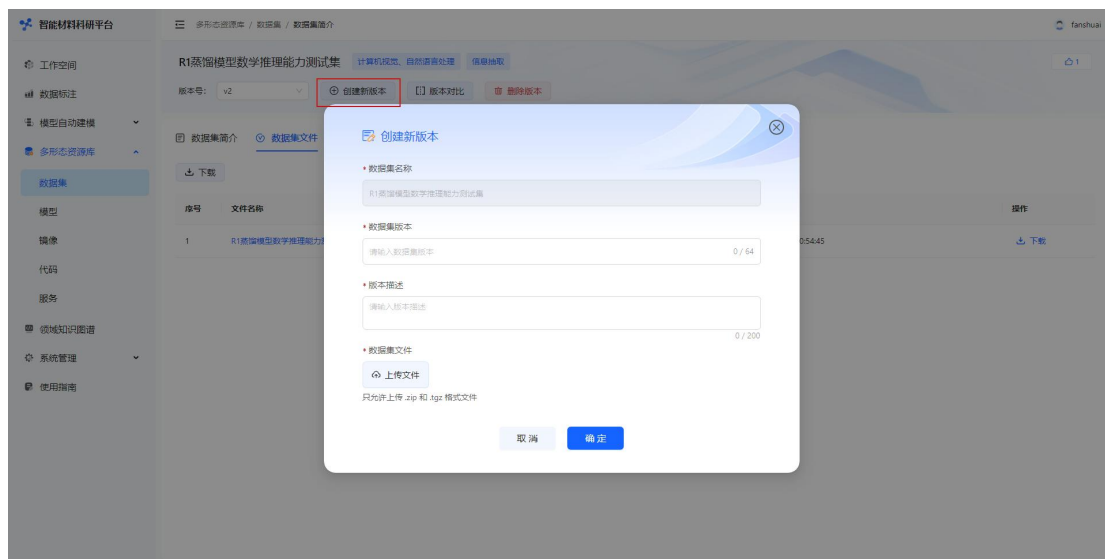


图 2-94 创建数据集新版本

## 2.9.5 版本对比

选择两个版本的数据集进行版本对比，比较数据集信息。

## 步骤:

1. 点击【数据集卡片】跳转到数据集简介界面。
2. 点击【版本对比】按钮弹出【请选择版本】弹窗，如图 2-95 所示。
3. 点击选择两个版本后点击【确定】按钮，弹出【数据集版本对比】弹窗，显示版本差异对比结果，如图 2-96 所示。

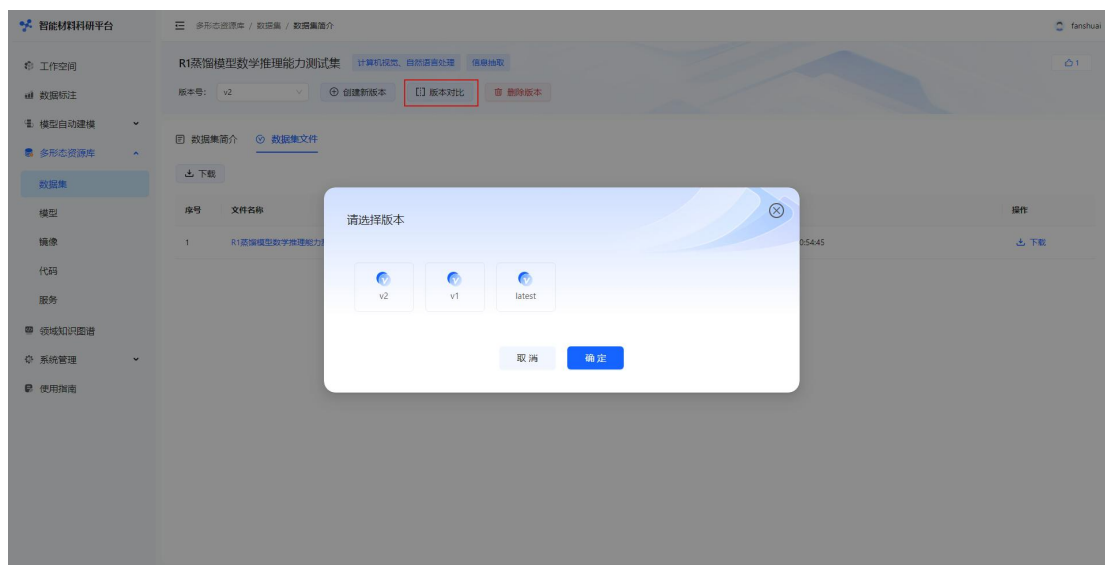


图 2-95 请选择数据集版本



图 2-96 数据集版本对比

## 2.9.6 下载数据集

将所需数据集下载到本地，可下载单个文件，也可打包全部文件下载。

## 步骤:

1. 点击【数据集卡片】跳转到数据集简介界面，点击【版本号】下拉框点击下载数据集文件版本号切换成功。
2. 点击【数据集文件】显示数据集版本的文件列表。
3. 点击列表上方的【下载】按钮，即可打包下载整个版本数据集的全部文件，如图 2-97 所示。
4. 点击数据集某个文件的【文件名称】或操作列【下载】按钮，即可下载单个该数据集文件，如图 2-98 所示。

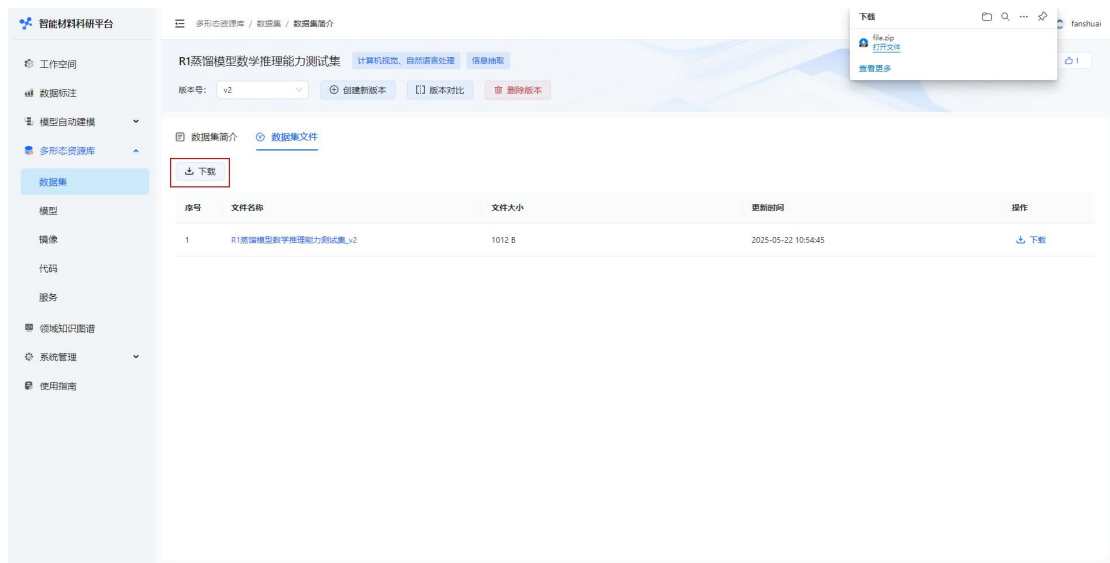


图 2-97 打包全部数据集文件下载

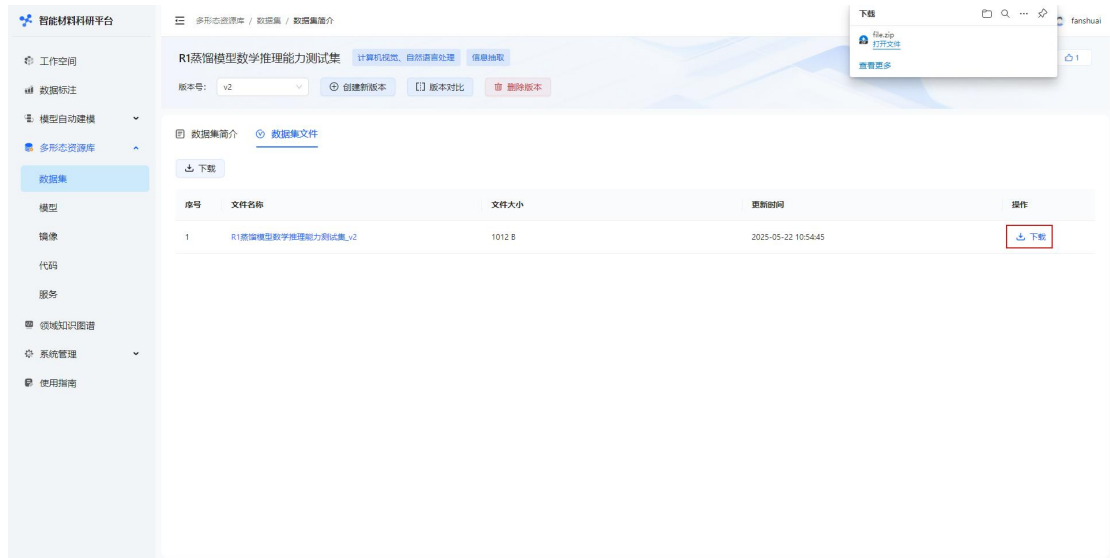


图 2-98 单个数据集文件下载

## 2.9.7 删除版本

选择不再需要的数据集版本进行删除操作，释放存储空间，同时系统会确保

删除操作的安全性，避免误删重要数据。

### 步骤：

1. 点击【数据集卡片】跳转到数据集简介界面，点击【版本号】下拉框点击要删除的版本号切换成功。

2. 点击【删除版本】-【确认】按钮，如图 2-99 所示。如果当前数据集只有一个版本，删除失败弹框提示不能删除该版本；如果存在多个版本，则删除版本成功。如果数据集版本无流水线模板使用，则成功删除数据集版本，如果被流水线模板使用则删除失败且弹框提示。

3. 点击【模型自动建模】-【流水线模板】菜单进入流水线模板模块，流水线模板列表中删除数据集版本被使用的流水线模板，参考章节 2.5.7。

4. 返回数据集简介界面再次点击【删除版本】-【确认】按钮，成功删除数据集版本。

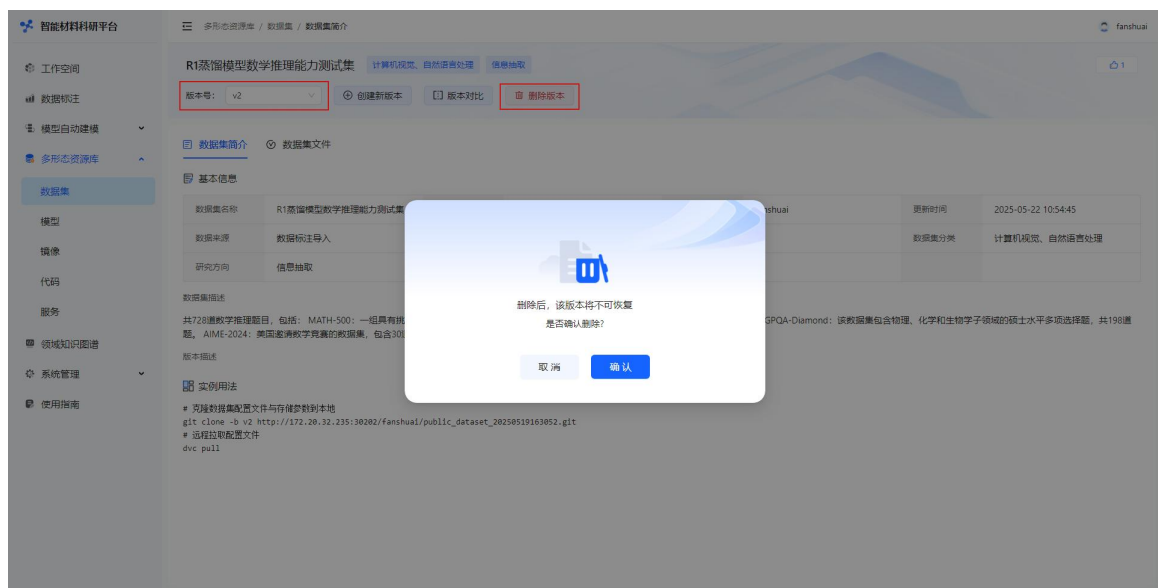


图 2-99 删除版本

## 2.9.8 删除数据集

选择不再需要的数据集进行删除操作，释放存储空间，同时系统会确保删除操作的安全性，避免误删重要数据。

### 步骤：

1. 进入【个人数据】界面，点击要删除数据集卡片的【删除图标】-【确认】按钮，如图 2-100 所示。如果数据集无流水线模板使用，则成功删除数据集，如

果被流水线模板使用则删除失败且弹框提示。

2. 点击【模型自动建模】-【流水线模板】菜单进入流水线模板模块，流水线模板列表中删除数据集被使用的流水线模板，参考章节 2.5.7。

3. 返回【个人数据】界面再次点击【删除图标】-【确认】按钮，成功删除数据集。

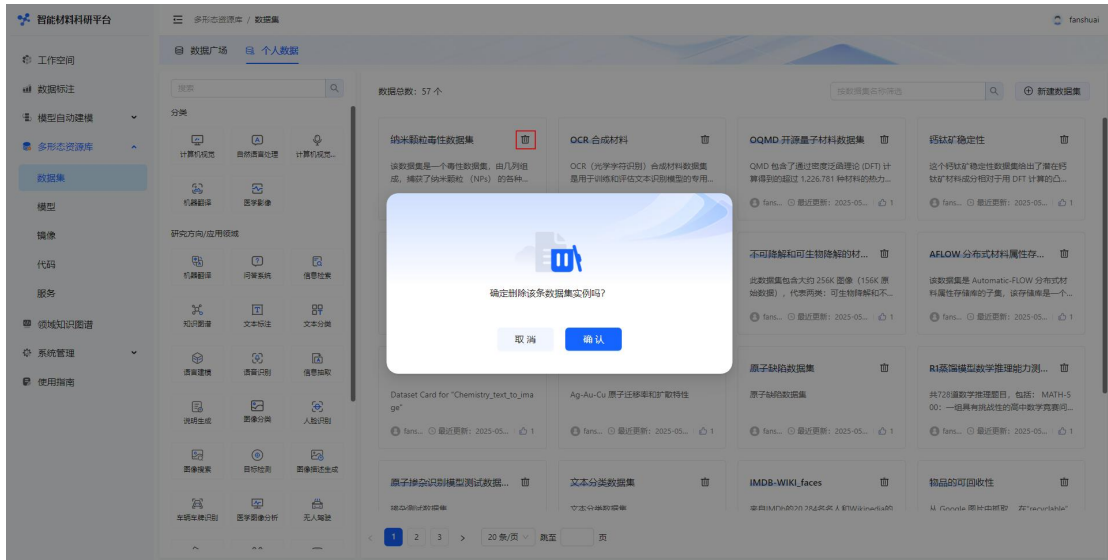


图 2-100 删除数据集

## 2.10 模型管理

模型管理包含模型广场和个人模型，模型广场汇聚了常用的经典的开源模型，个人模型则是用户自己上传的模型。此功能是实验功能及服务管理功能的基础。实验功能依赖于模型功能。用户选择了对应的模型，即可在此模型上进行相应的流水线自动实验，以及实验后导出新的结果模型。也可以将模型部署成持久服务，提供用户访问的地址对模型服务进行访问。

### 2.10.1 查看模型管理模块

介绍用户如何进入模型管理模块界面，分为公开模型（模型广场）和个人私有模型，模型卡片根据点赞数排序推荐，可通过关键词、标签等条件快速检索目标模型。

#### 步骤：

1. 1.点击菜单栏【多形态资源库】-【模型】菜单进入模型广场界面，如图 2-101 所示。

2. 点击侧边栏【模型框架】或【模型能力】标签，界面显示对应标签的模型。

3. 模型名称筛选框输入关键字，按 Enter 键或点击搜索图标将进行模糊搜索，界面只显示模型名称包括关键字的模型，如图 2-102 所示。



图 2-101 模型管理模块

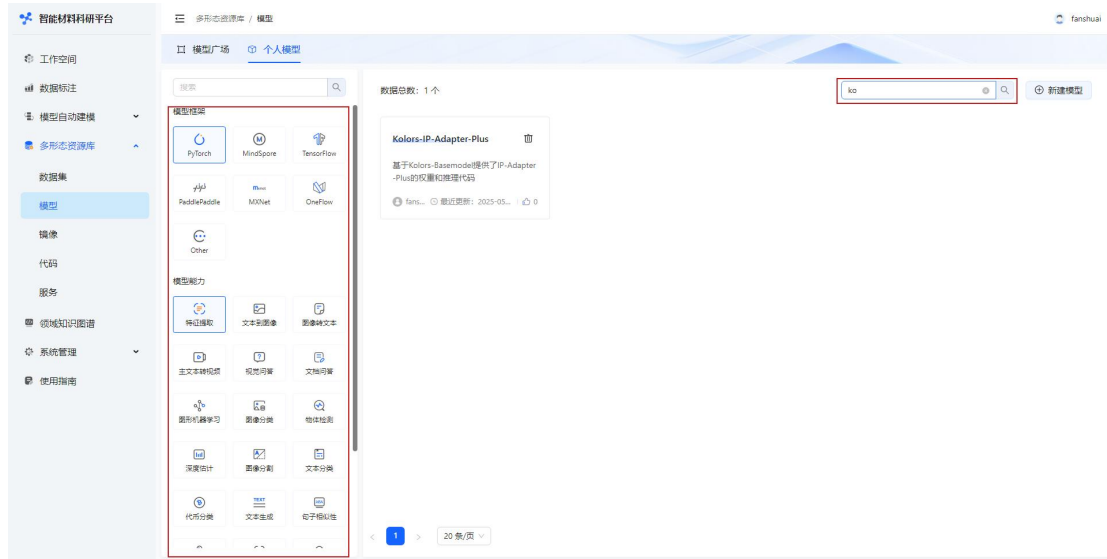


图 2-102 查询模型

### 2.10.2 新建模型

将新模型添加到平台，并为其添加描述、标签等信息，便于后续管理和使用。

步骤:

1. 点击【个人模型】-【新建模型】按钮弹出【新建模型】弹窗，如图 2-103 所示。

2. 输入模型名称、模型版本、模型描述、版本描述，选择可见性和上传模型文件，选填模型框架和模型能力，点击弹窗【确定】按钮成功新建模型。



图 2-103 新建模型

### 2.10.3 查看模型

选择数模型版本后，查看模型的详细版本信息，包括版本号、更新时间、训练参数等元数据。

步骤：

1. 点击【模型卡片】跳转到模型简介界面，用户可查看到模型的名称，版本，创建人，更新时间等基本信息和指标参数差异对比，如图 2-104 所示。
2. 点击【模型文件】显示模型版本的文件和文件大小等内容，如图 2-105 所示。
3. 点击【模型演化】可查看模型不同版本的演化信息，如图 2-106 所示。
4. 点击【版本号】下拉框可选择切换模型不同版本查看。



图 2-104 模型简介

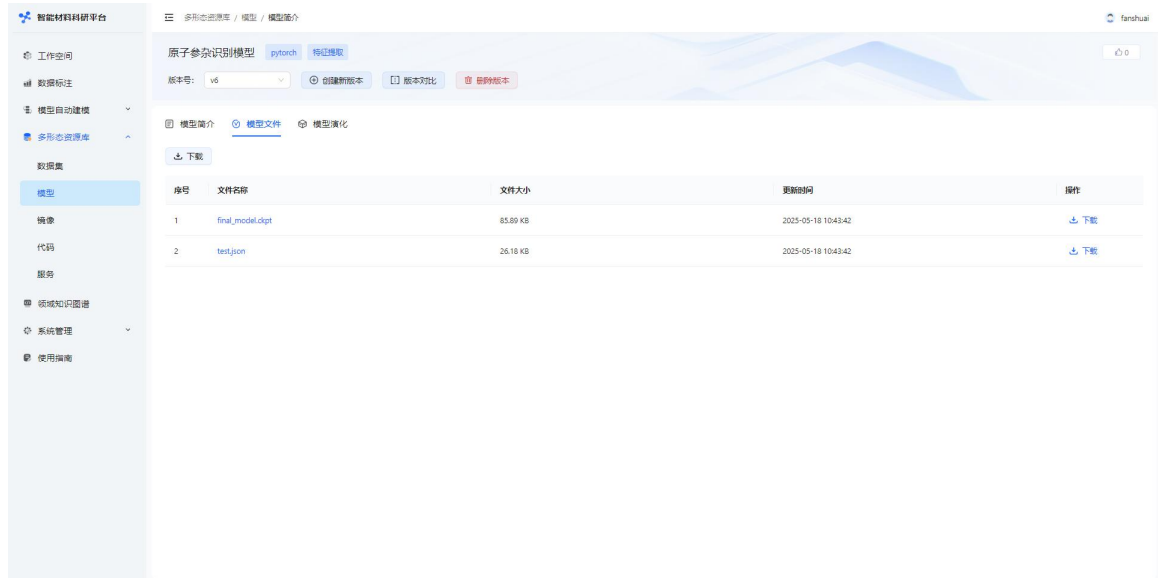


图 2-105 模型文件



图 2-106 模型演化

## 2.10.4 创建新版本

创建模型新版本实现模型迭代的精准管控与全生命周期追溯。

步骤：

1. 点击【模型卡片】跳转到模型简介界面。
2. 点击【创建新版本】按钮弹出创建新版本弹窗，如图 2-107 所示。
3. 输入模型版本和版本描述，上传模型文件，点击弹窗【确定】按钮成功创建新版本。

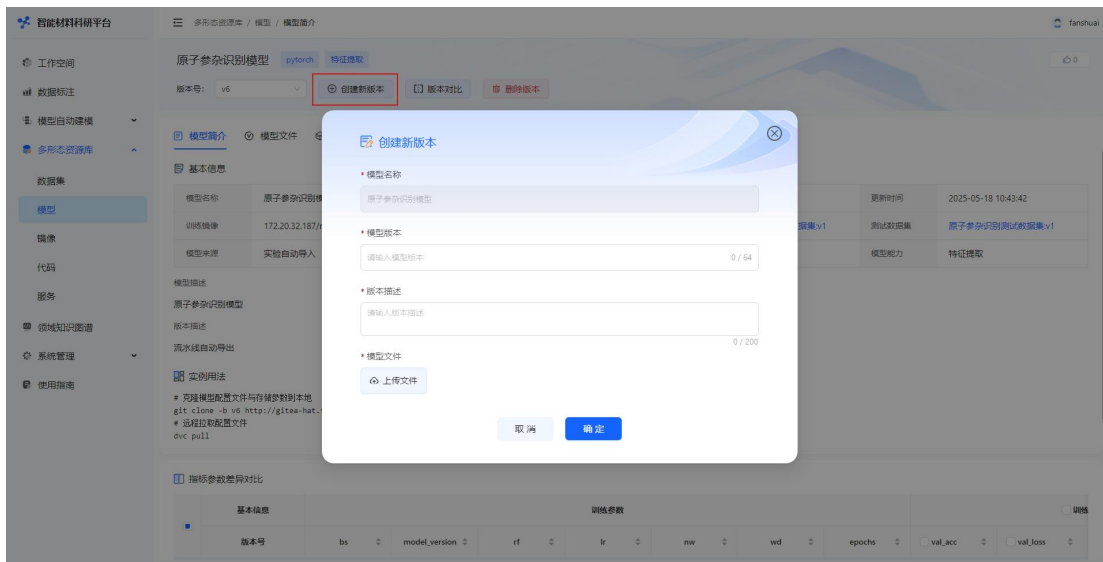


图 2-107 创建模型新版本

## 2.10.5 版本对比

选择两个版本的模型进行版本对比，比较模型信息。

## 步骤:

1. 点击【模型卡片】跳转到模型简介界面。
2. 点击【版本对比】按钮弹出【请选择版本】弹窗，如图 2-108 所示。
3. 点击选择两个版本后点击【确定】按钮，弹出【模型版本对比】弹窗，显示版本差异对比结果，如图 2-109 所示。

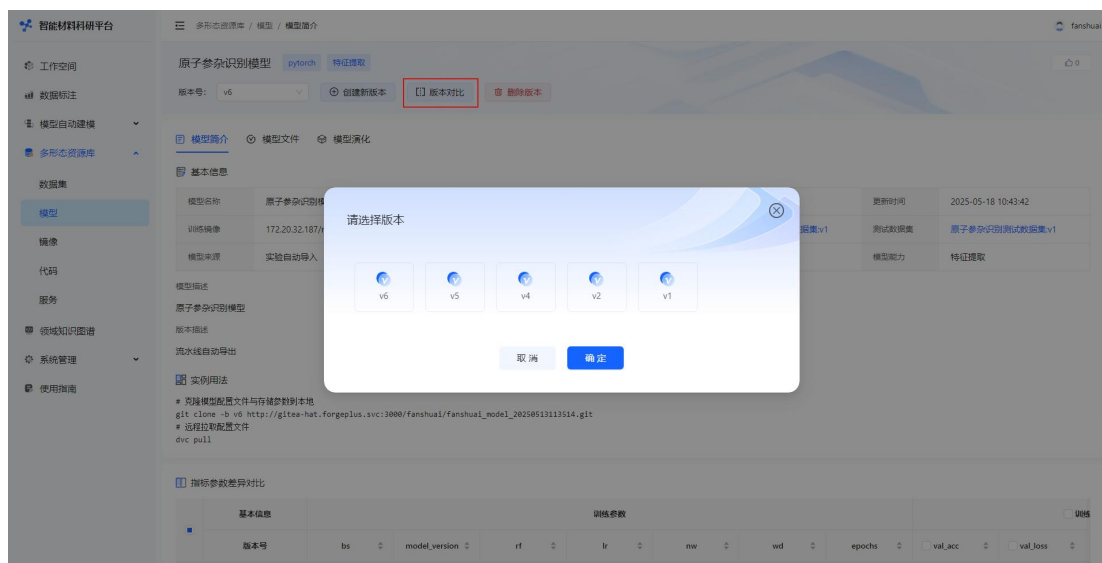


图 2-108 请选择模型版本

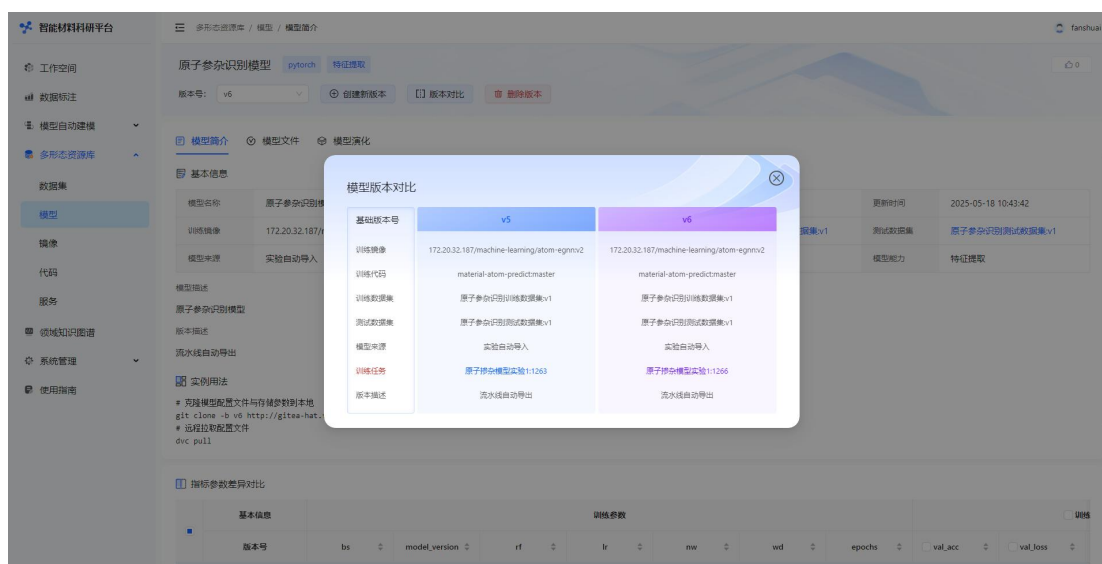


图 2-109 模型版本对比

## 2.10.6 下载模型

将所需模型下载到本地，可下载单个文件，也可打包全部文件下载。

## 步骤:

1. 点击【模型卡片】跳转到模型简介界面，点击【版本号】下拉框点击下载模型文件版本号切换成功。
2. 点击【模型文件】显示模型版本的文件列表。
3. 点击列表上方的【下载】按钮，即可打包下载整个版本模型的全部文件，如图 2-110 所示。
4. 点击模型某个文件的【文件名称】或操作列【下载】按钮，即可下载单个该模型文件，如图 2-111 所示。

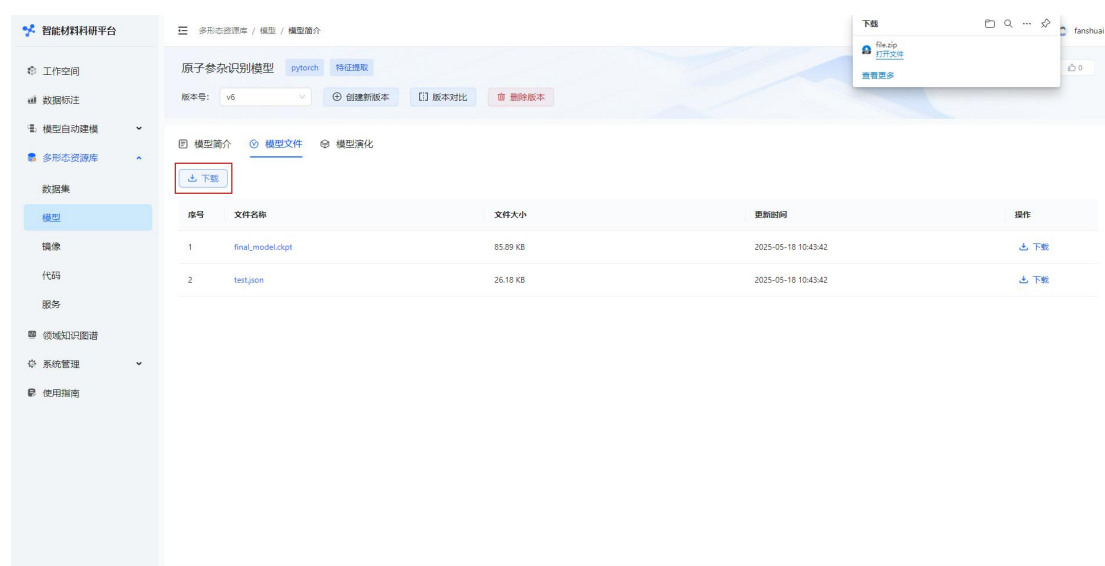


图 2-110 打包全部模型文件下载

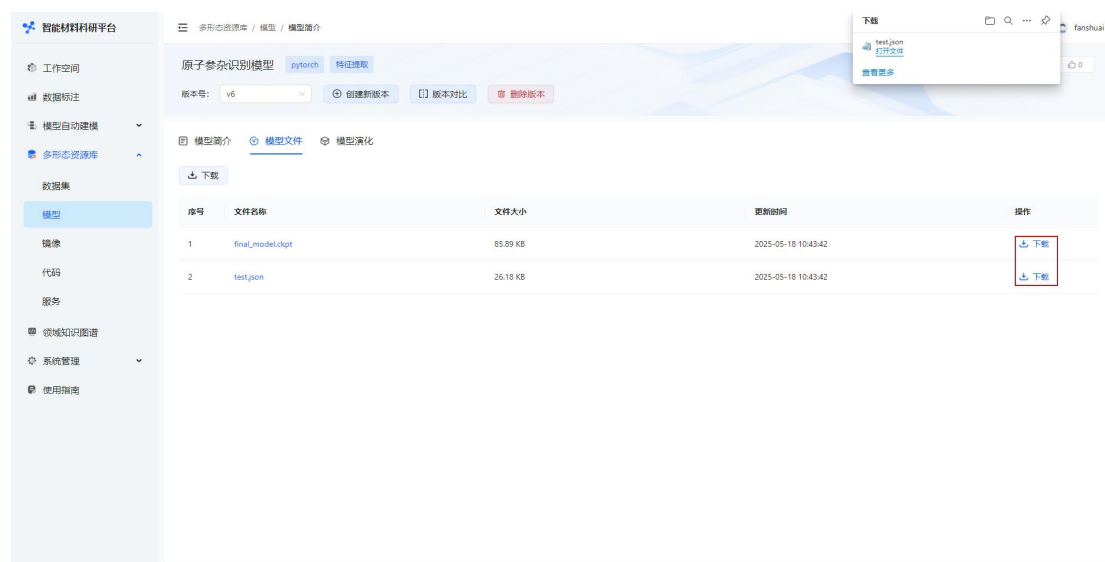


图 2-111 单个模型文件下载

## 2.10.7 删除版本

选择不再需要的模型版本进行删除操作，释放存储空间，同时系统会确保删

除操作的安全性，避免误删重要数据。

### 步骤：

1. 点击【模型卡片】跳转到模型简介界面，点击【版本号】下拉框点击要删除的版本号切换成功。

2. 点击【删除版本】-【确认】按钮，如图 2-112 所示。如果当前模型只有一个版本，删除失败弹框提示不能删除该版本；如果存在多个版本，则删除版本成功。如果模型版本无流水线模板使用，则成功删除模型版本，如果被流水线模板使用则删除失败且弹框提示。

3. 点击【模型自动建模】-【流水线模板】菜单进入流水线模板模块，流水线模板列表中删除模型版本被使用的流水线模板，参考章节 2.5.7。

4. 返回模型简介界面再次点击【删除版本】-【确认】按钮，成功删除模型版本。

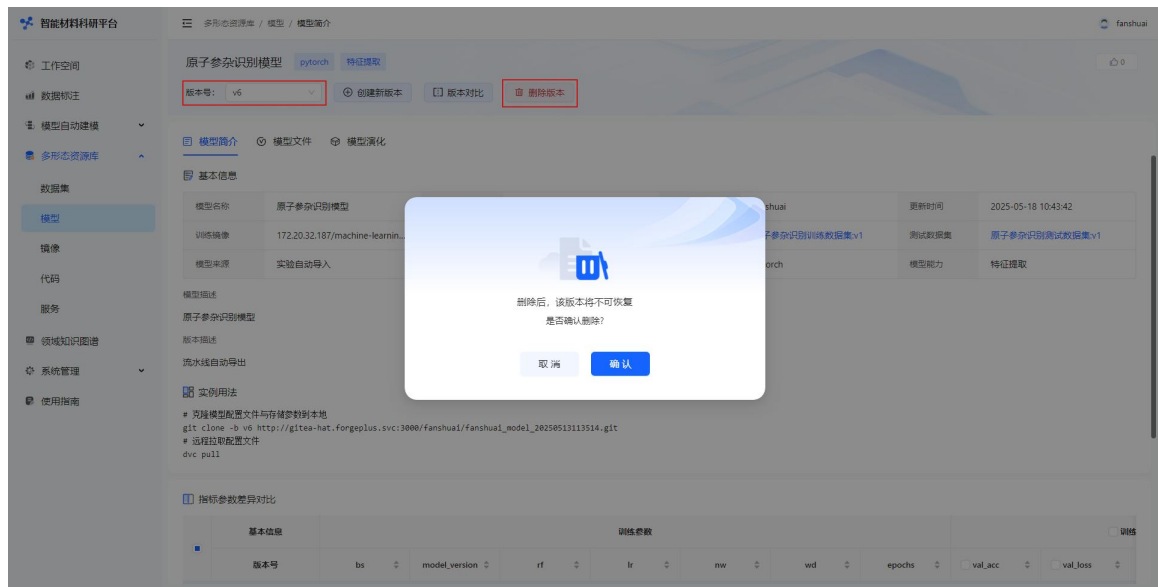


图 2-112 删除模型版本

## 2.10.8 删除模型

选择不再需要的模型进行删除操作，释放存储空间，同时系统会确保删除操作的安全性，避免误删重要数据。

### 步骤：

1. 进入【个人数据】界面，点击要删除模型卡片的【删除图标】-【确认】按钮，如图 2-113 所示。如果模型无流水线模板使用，则成功删除模型，如果被

流水线模板使用则删除失败且弹框提示。

2. 点击【模型自动建模】-【流水线模板】菜单进入流水线模板模块，流水线模板列表中删除模型被使用的流水线模板，参考章节 2.5.7。

3. 返回【个人数据】界面再次点击【删除图标】-【确认】按钮，成功删除模型。

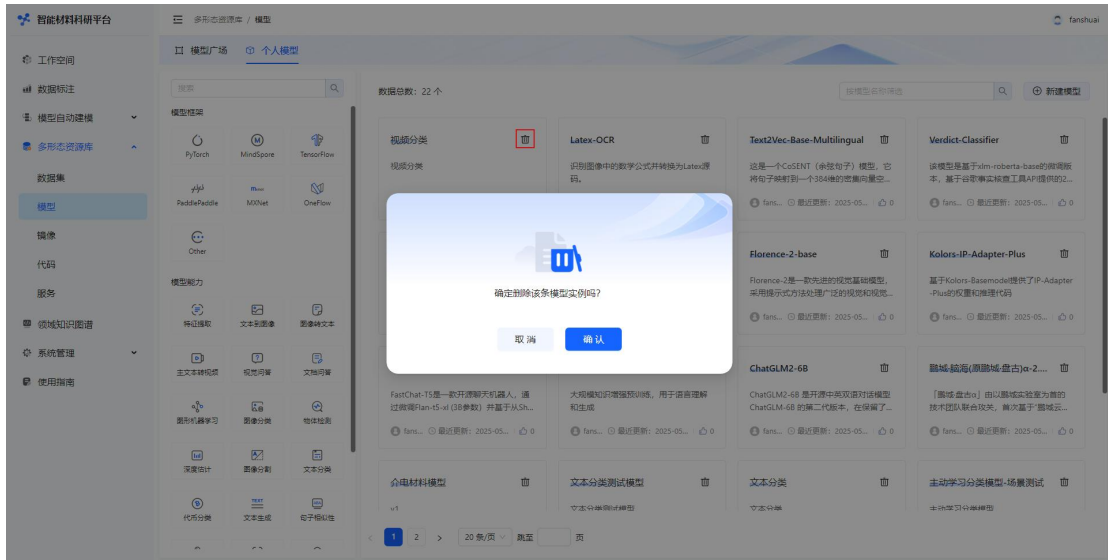


图 2-113 删除模型

## 2.11 镜像管理

镜像管理提供了组件运行需要的镜像和各深度学习框架的镜像，同时提供制作镜像的功能，用户可以上传自己的镜像到镜像仓库中。

### 2.11.1 查看镜像管理模块

介绍用户如何进入镜像管理模块界面，分为公开镜像和个人镜像，可通过关键词快速检索目标镜像。

**步骤：**

1. 点击菜单栏【多形态资源库】-【镜像】菜单进入公共镜像界面，如图 2-114 所示。

2. 镜像名称筛选框输入关键字，按 Enter 键或点击搜索图标将进行模糊搜索，界面只显示镜像名称包括关键字的镜像，如图 2-115 所示。



图 2-114 镜像管理模块

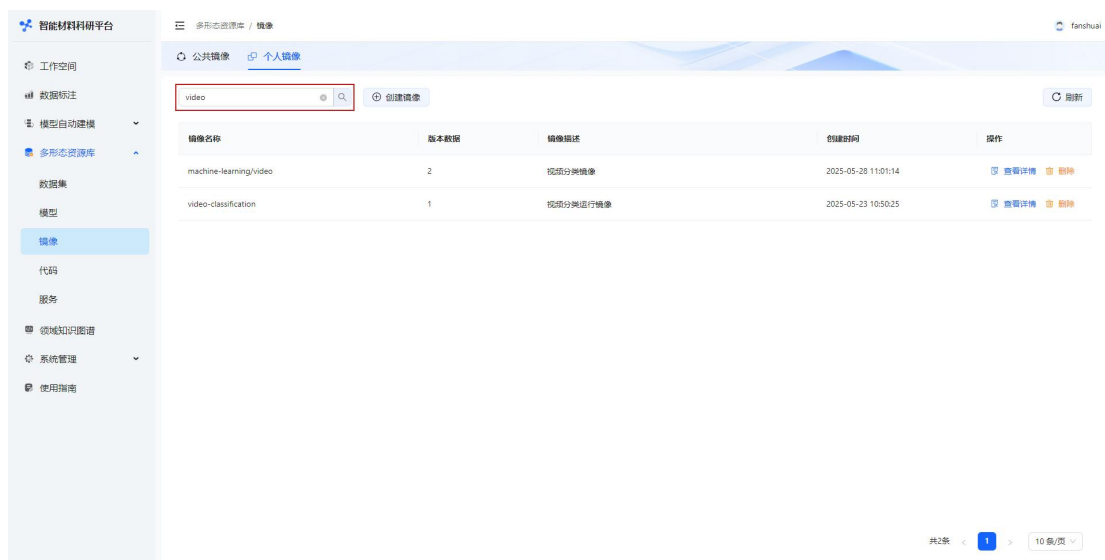


图 2-115 查询镜像

### 2.11.2 创建镜像

创建个人镜像，镜像制作可以通过上传 tar 包方式和从公网镜像仓库导入两种方式。

**步骤：**

1. 点击【个人镜像】-【创建镜像】按钮跳转到创建镜像页面，如图 2-116 所示。

2. 输入基本信息镜像名称和版本、镜像描述，镜像构建方式分为【基于公网镜像】和【本地上传】，选择【基于公网镜像】则输入镜像地址，选择【本地上传】则选择上传镜像文件。

3. 点击【确定】按钮创建镜像成功并开始构建镜像。

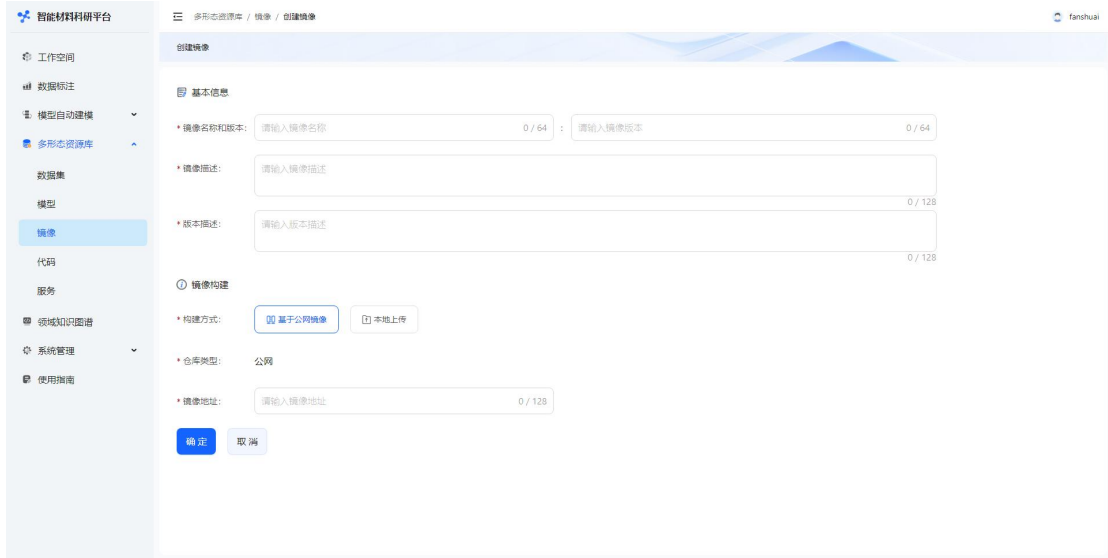


图 2-116 创建镜像

### 2.11.3 查看详情

查看镜像的基本信息和镜像版本。

步骤：

点击镜像列表操作【查看详情】按钮跳转到镜像详情页面，如图 2-117 所示。

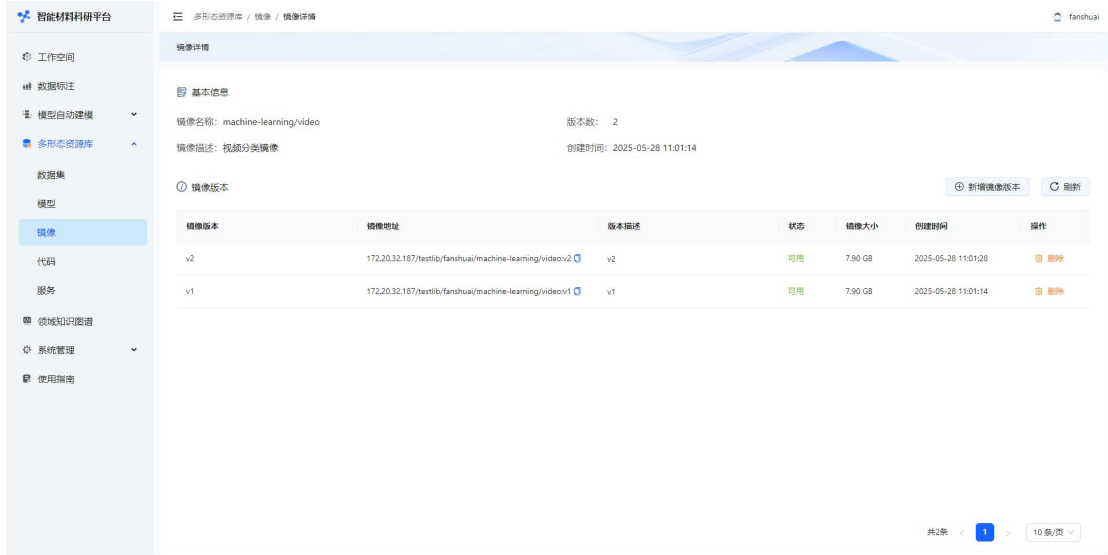


图 2-117 个人镜像详情

### 2.11.4 新增镜像版本

将新的个人镜像版本文件添加到平台。

步骤：

1. 点击个人镜像列表操作【查看详情】按钮跳转到镜像详情页面。

2. 点击镜像详情页面的【新增镜像版本】按钮跳转到新增镜像版本页面，如图 2-118 所示。

3. 输入基本信息镜像版本和版本描述，镜像构建方式分为【基于公网镜像】和【本地上传】，选择【基于公网镜像】则输入镜像地址，选择【本地上传】则选择上传镜像文件。

4. 点击【确定】按钮新增镜像版本成功并开始构建镜像。

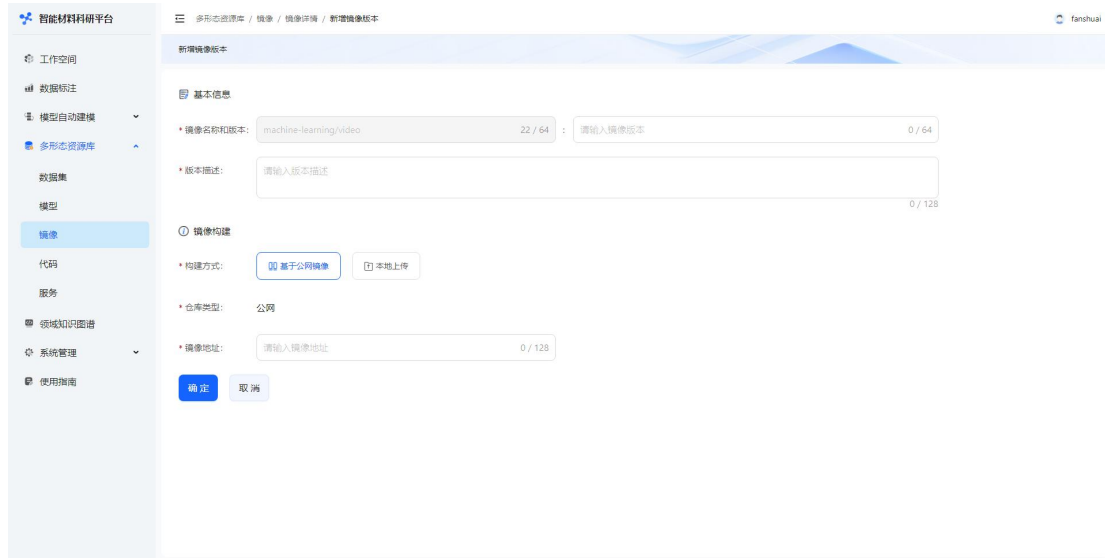


图 2-118 新增镜像版本

### 2.11.5 删除版本

选择不再需要的个人镜像版本进行删除操作，释放存储空间，同时系统会确保删除操作的安全性，避免误删重要数据。

#### 步骤：

1. 点击个人镜像列表操作【查看详情】按钮跳转到镜像详情页面。
2. 点击操作【删除】-【确认】按钮，如图 2-119 所示。如果当前镜像只有一个版本，删除失败弹框提示不能删除该版本；如果存在多个版本，则删除版本成功。如果镜像版本无流水线模板使用，则成功删除镜像版本，如果被流水线模板使用则删除失败且弹框提示。
3. 点击【模型自动建模】-【流水线模板】菜单进入流水线模板模块，流水线模板列表中删除镜像版本被使用的流水线模板，参考章节 2.5.7。
4. 返回镜像详情界面再次点击【删除】-【确认】按钮，成功删除镜像版本。

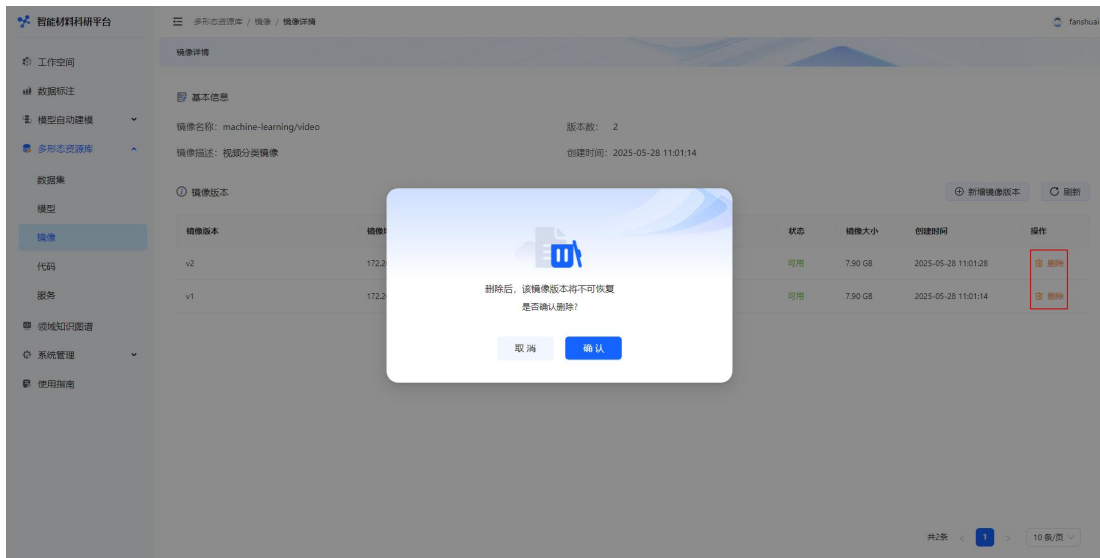


图 2-119 删除版本

## 2.11.6 删除镜像

选择不再需要的个人镜像进行删除操作，释放存储空间，同时系统会确保删除操作的安全性，避免误删重要数据。

### 步骤：

1. 进入【个人镜像】界面，点击列表要删除镜像操作【删除】-【确认】按钮，如图 2-120 所示。如果镜像无流水线模板使用，则成功删除镜像，如果被流水线模板使用则删除失败且弹框提示。

2. 点击【模型自动建模】-【流水线模板】菜单进入流水线模板模块，流水线模板列表中删除镜像被使用的流水线模板，参考章节 2.5.7。

3. 返回【个人镜像】界面再次点击【删除】-【确认】按钮，成功删除镜像。

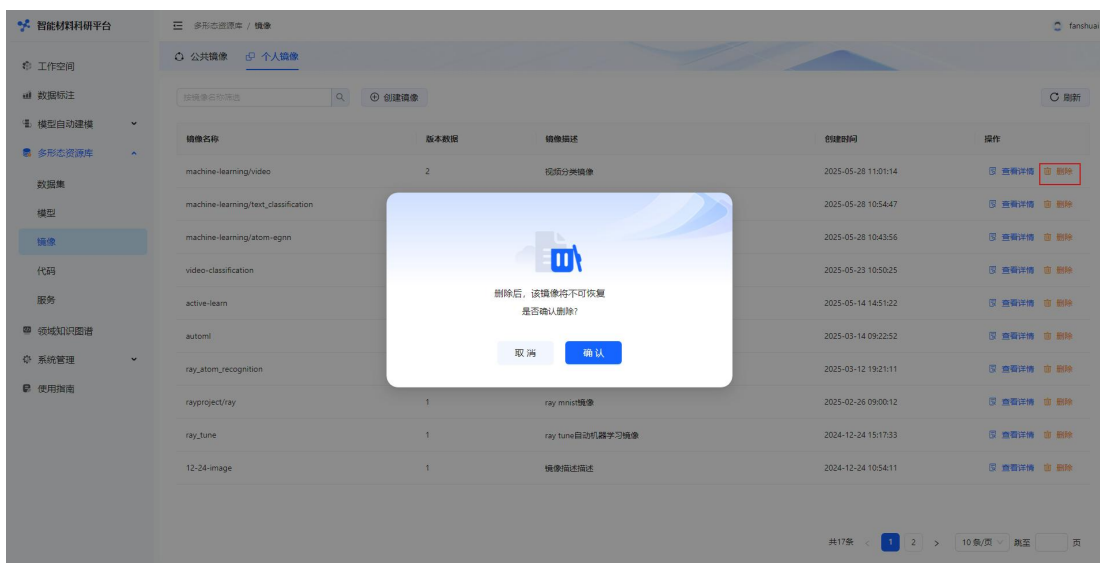


图 2-120 删除镜像

## 2.12 代码管理

代码管理是实验功能的基础，在实验中，用户可以选择代码配置，自动的完成代码从 git 的挂载。

### 2.12.1 查看代码管理模块

介绍用户如何进入代码管理模块界面，可通过关键词快速检索目标代码配置。

步骤：

1. 点击菜单栏【多形态资源库】-【代码】菜单进入代码配置界面，如图 2-121 所示。
2. 代码仓库名称筛选框输入关键字，按 **Enter** 键或点击搜索图标将进行模糊搜索，界面只显示代码仓库名称包括关键字的代码配置，如图 2-122 所示。

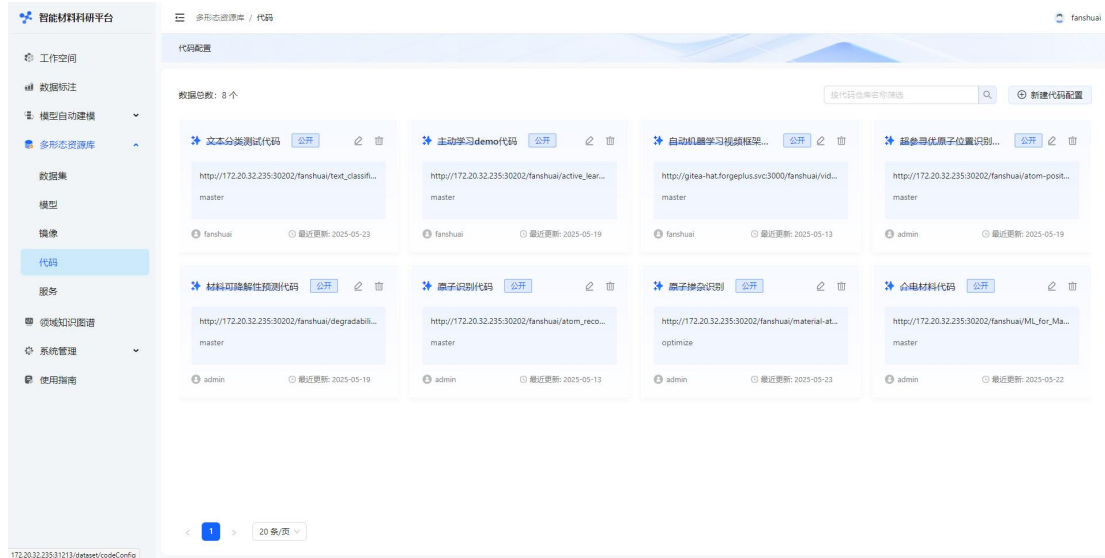


图 2-121 代码管理模块

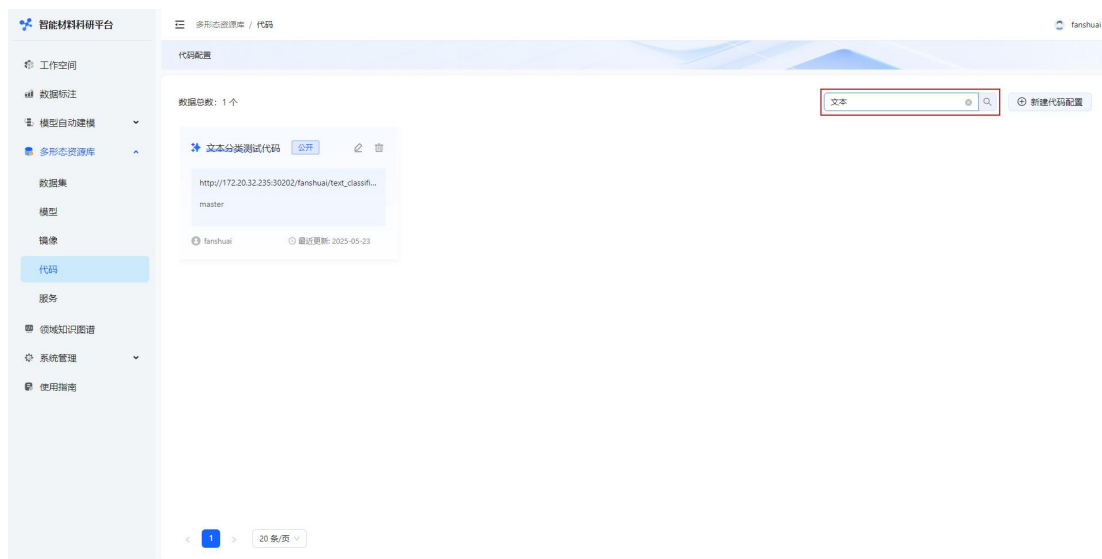


图 2-122 查询代码配置

## 2.12.2 新建代码配置

新增代码配置，输入代码地址，分支，若是私有仓库则需要输入用户名密码。

### 步骤：

1. 代码管理模块界面点击【新建代码配置】按钮弹出【新建代码配置】弹窗，如图 2-123 所示。
2. 输入代码仓库名称，选择代码仓库可见性，输入 Git 地址和代码分支/Tag。如果代码仓库可见性选择私有，需选择验证方式为用户名/密码或 SSH Key，验证方式选择用户名/密码需输入 Git 用户名和 Git 密码，如图 2-123 所示，验证方式选择 SSH Key 需输入 SSH Key。

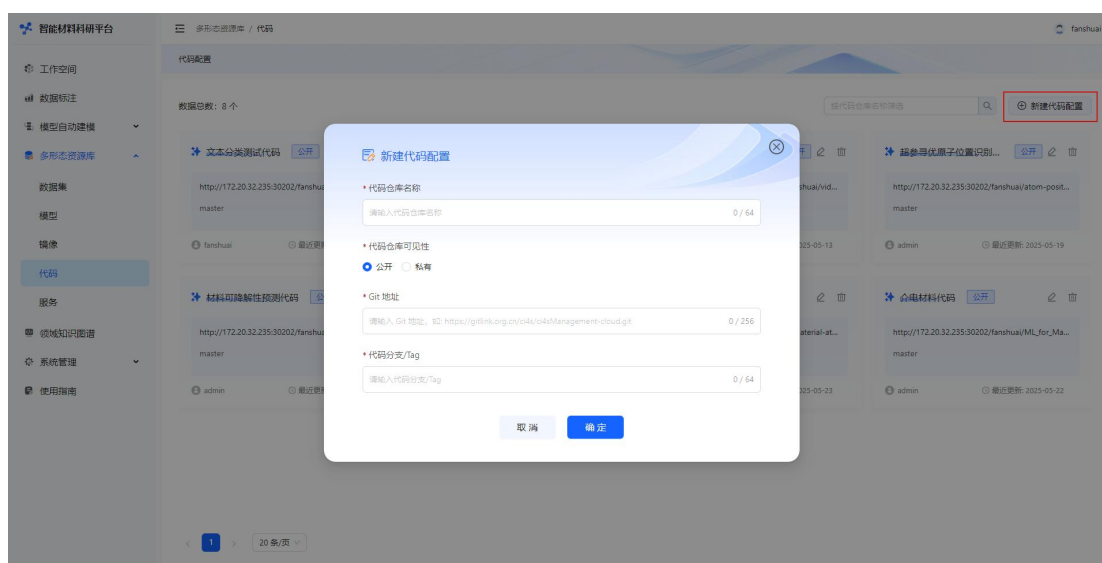


图 2-123 新建公开代码配置

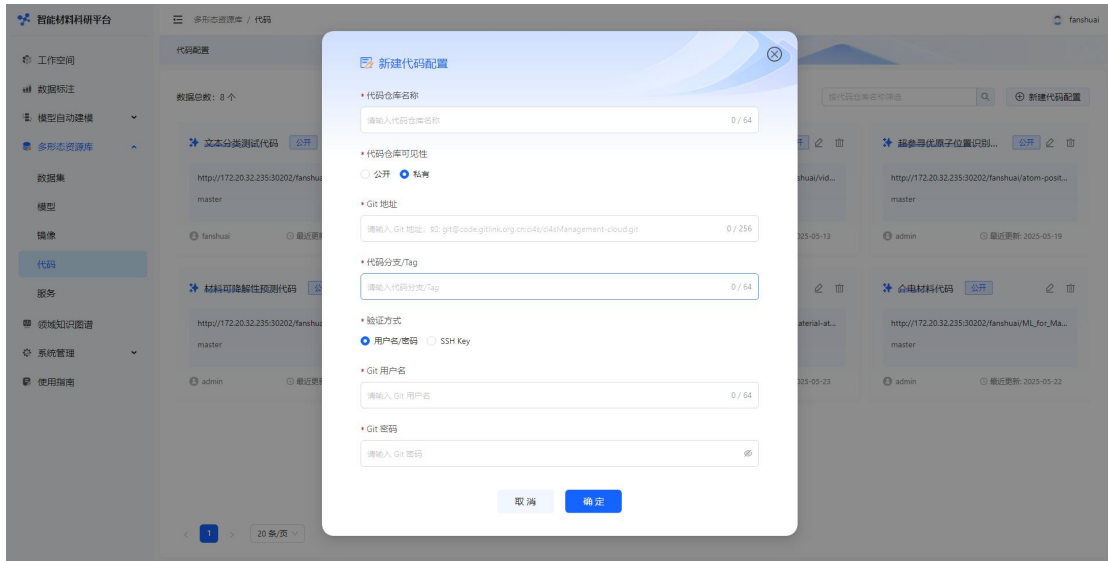


图 2-124 新建私有代码配置

### 2.12.3 修改代码配置

编辑相关配置信息修改代码配置。

步骤：

1. 代码管理模块界面点击代码卡片的【修改图标】弹出【修改代码配置】弹窗，如图 2-125 所示。
2. 修改代码相关配置后点击弹窗【确定】按钮保存成功。

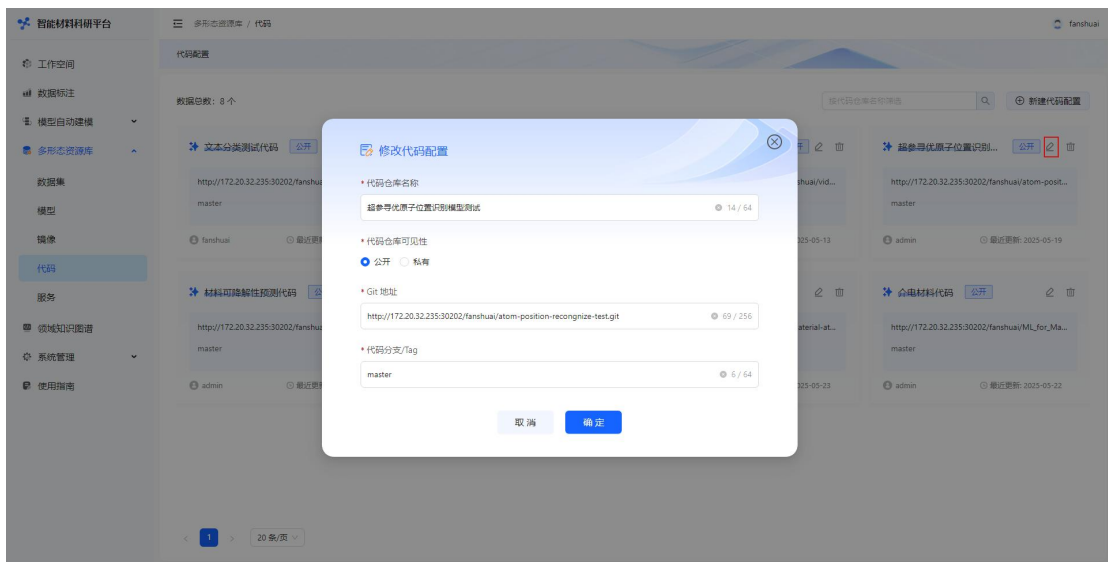


图 2-125 修改代码配置

### 2.12.4 删除代码配置

选择不再需要的代码配置进行删除操作，释放存储空间，同时系统会确保删除操作的安全性，避免误删重要配置。

### 步骤:

1. 代码管理模块界面点击代码卡片的【删除图标】-【确认】按钮,如图 2-126 所示。如果代码配置无流水线模板使用,则成功删除代码配置,如果被流水线模板使用则删除失败且弹框提示。

2. 点击【模型自动建模】-【流水线模板】菜单进入流水线模板模块,流水线模板列表中删除代码配置被使用的流水线模板,参考章节 2.5.7。

3. 返回代码管理模块界面再次点击代码卡片的【删除图标】-【确认】按钮,成功删除代码配置。

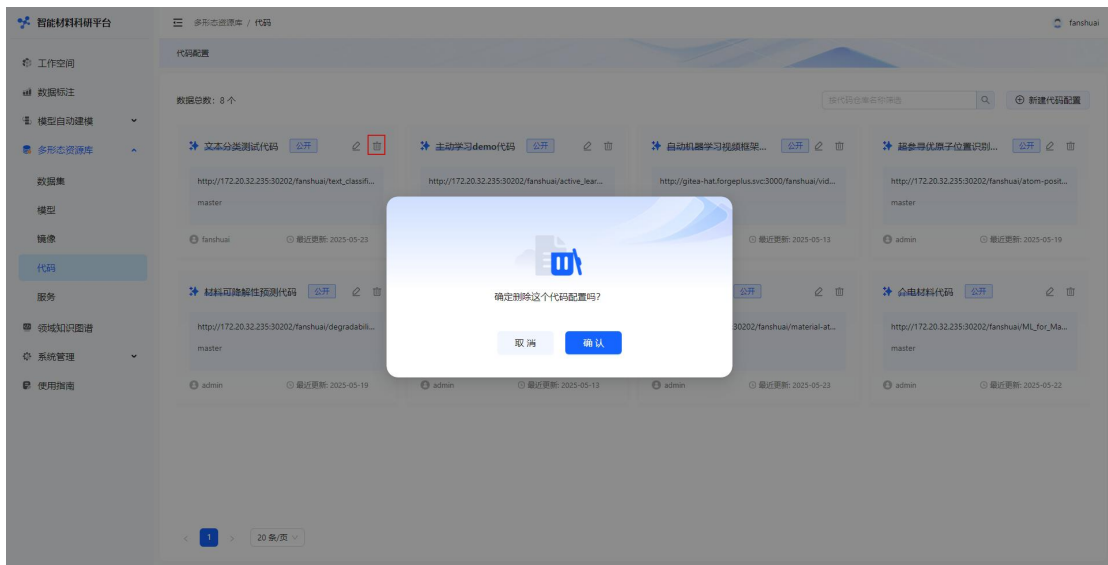


图 2-126 删除代码配置

## 2.13 服务管理

服务管理提供智能模型服务化部署功能,支持文本、图片、视频等多种输入输出格式的智能模型服务部署,主要功能是对服务进行启停管理,版本控制管理等。

### 2.13.1 查看服务管理模块

介绍用户如何进入服务管理模块界面,可通过关键词快速检索目标服务。

### 步骤:

1. 点击菜单栏【多形态资源库】-【服务】菜单进入服务列表界面,如图 2-127 所示。

2. 服务名称筛选框输入关键字,服务类型下拉框可选择【全部、视频、图

片、音频或文本】，按 **Enter** 键或点击搜索图标将进行模糊搜索，界面只显示对应服务类型中服务名称包括关键字的服务，如图 2-128 所示。



图 2-127 服务管理模块界面

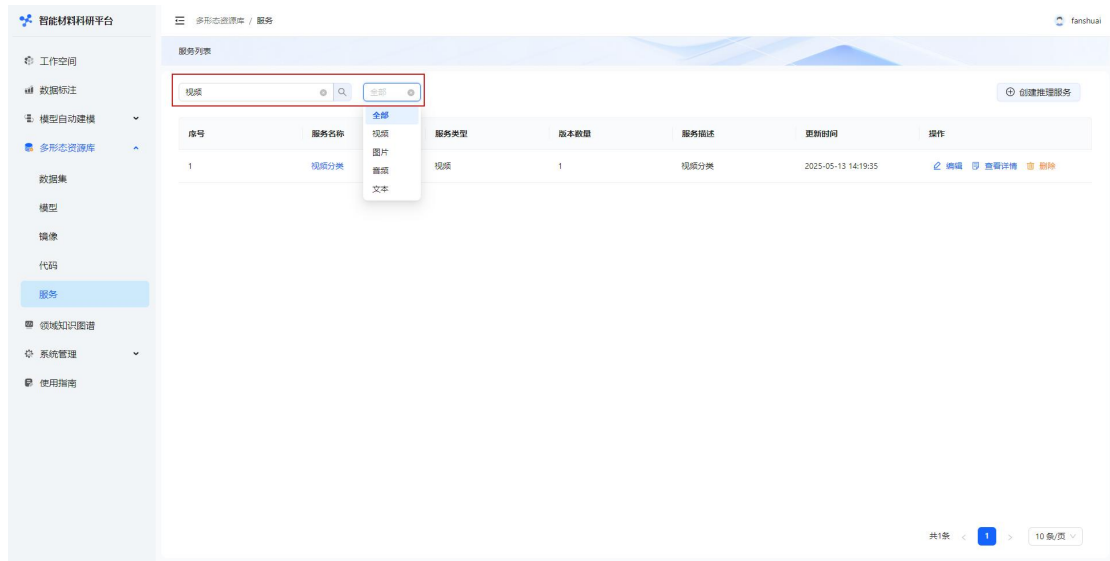


图 2-128 查询服务

### 2.13.2 创建推理服务

新增服务，填写服务名称等信息。

**步骤：**

1. 服务列表界面点击【创建推理服务】按钮跳转到【创建推理服务】界面，如图 2-129 所示。
2. 输入服务名称和描述，选择服务类型，点击【确定】按钮弹窗提示创建服务成功，用户可选择是否继续创建服务版本，如图 2-130 所示。

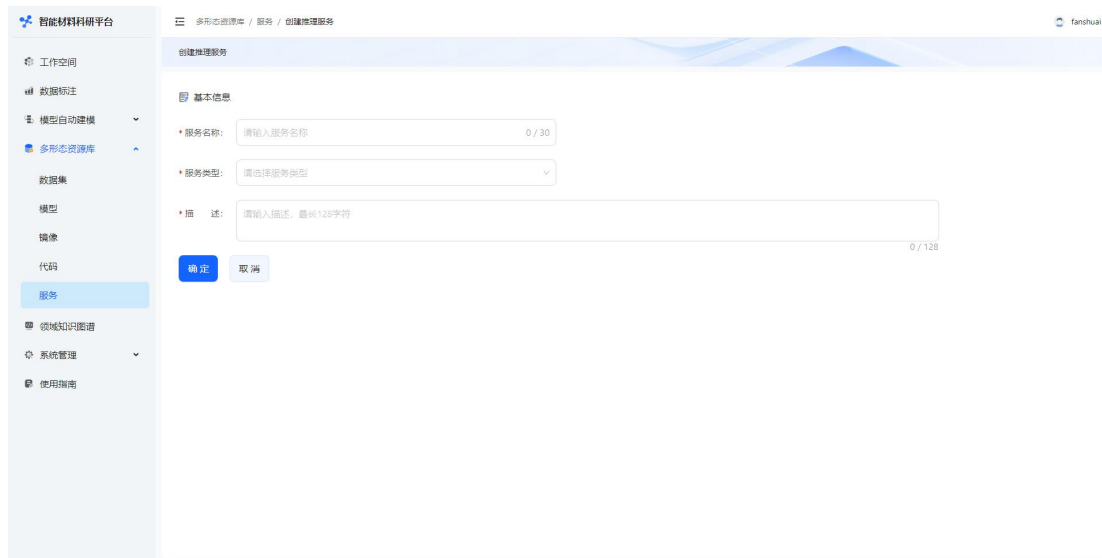


图 2-129 创建推理服务

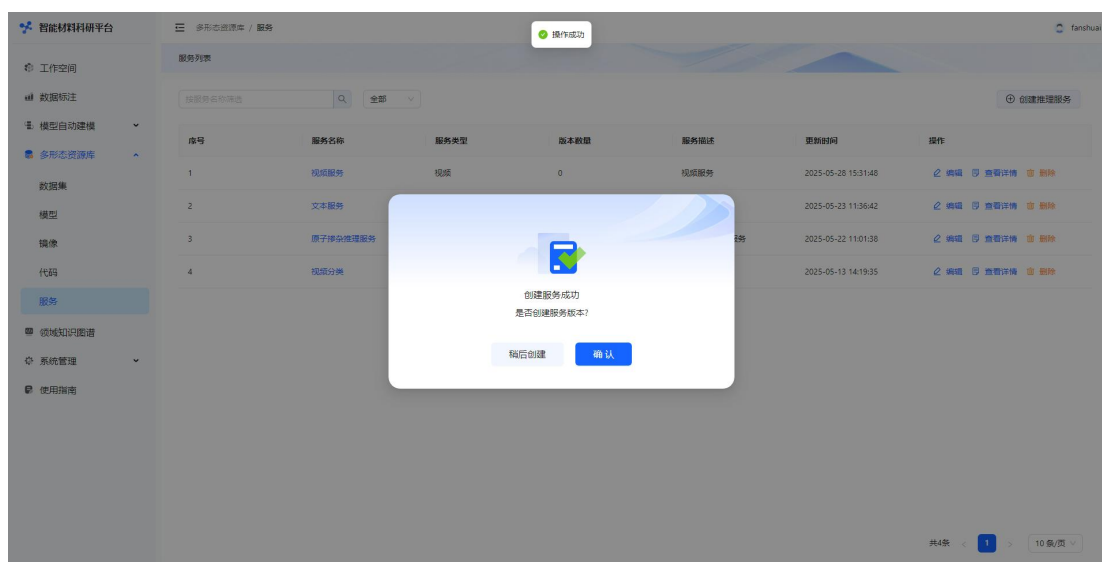


图 2-130 创建服务成功

### 2.13.3 编辑推理服务

编辑推理服务相关描述。

**步骤：**

1. 服务列表界面点击操作【编辑】按钮跳转到【编辑推理服务】界面，如图 2-131 所示。
2. 可编辑服务类型或描述，点击【确定】按钮成功编辑推理服务。

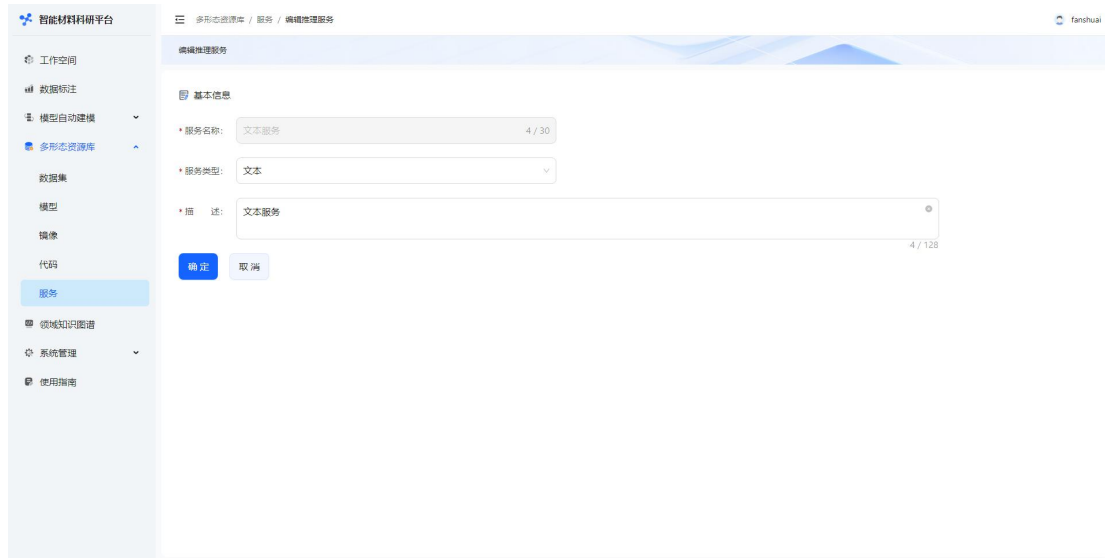


图 2-131 编辑推理服务

## 2.13.4 查看服务详情

查看服务基本信息和服务版本列表，展示服务的所有版本信息，包括版本号、部署状态等，便于用户查看和管理服务的历史版本。

**步骤：**

服务列表界面点击操作【查看详情】按钮跳转到【服务详情】界面，如图 2-132 所示。



图 2-132 服务详情

## 2.13.5 新增服务版本

将模型或应用一键部署为在线服务，支持配置资源、镜像、副本数、存储路径、环境变量等参数。

## 步骤:

1. 服务列表界面点击操作【查看详情】按钮跳转到【服务详情】界面。
2. 点击【新增服务版本】按钮跳转到【新增服务版本】界面，如图 2-133 所示。
3. 输入基本信息和设置部署构建相关配置，点击【新建】按钮成功新增服务版本。

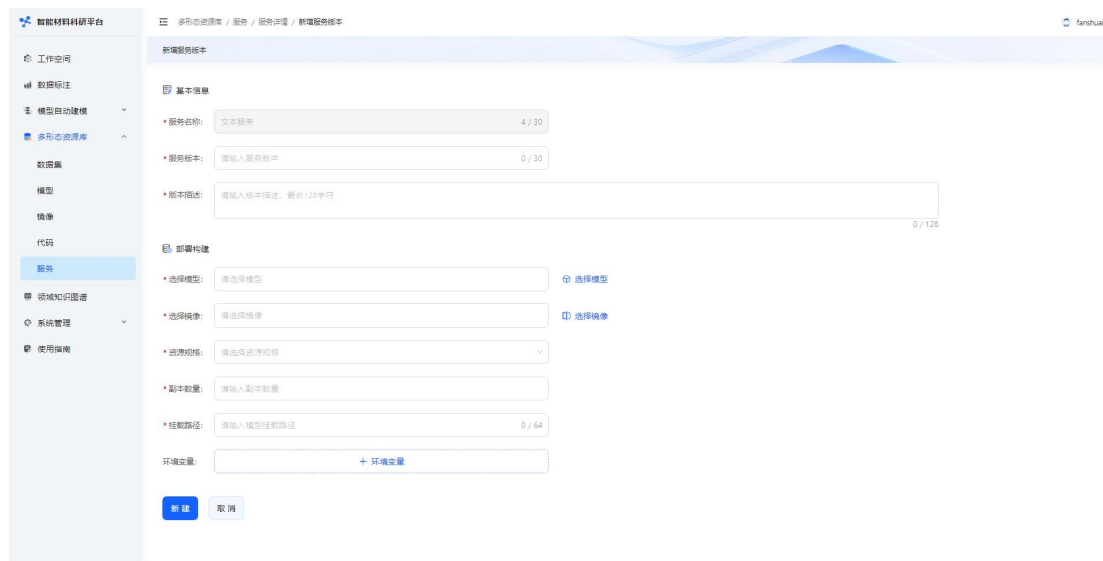


图 2-133 新增服务版本

### 2.13.6 更新服务版本

对已部署的服务进行版本更新资源、副本数，确保服务功能的持续优化和迭代，同时最小化对用户的影响。

## 步骤:

1. 服务列表界面点击操作【查看详情】按钮跳转到【服务详情】界面。
2. 服务版本列表点击操作【更新】按钮跳转到【更新服务版本】界面，如图 2-134 所示。
3. 可修改版本描述、资源规格和副本数量，点击【更新】按钮成功更新服务版本。

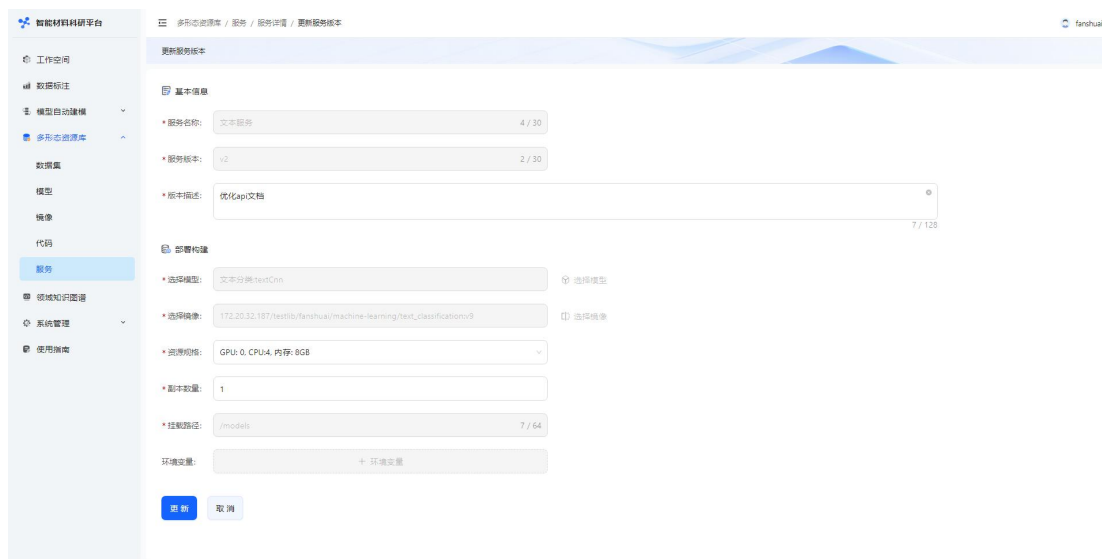


图 2-134 更新服务版本

## 2.13.7 服务启停

用户可以手动启动或停止服务，灵活控制服务的运行状态，节省资源并满足临时需求。

### 步骤：

1. 服务列表界面点击操作【查看详情】按钮跳转到【服务详情】界面。
2. 服务版本列表点击操作【重启】按钮跳转到【重启服务版本】界面，可修改版本描述、资源规格和副本数量，如图 2-135 所示，点击【重启】按钮，服务先挂起后运行。
3. 服务版本列表点击操作【停止】-【确认】按钮成功停止服务，如图 2-136 所示。

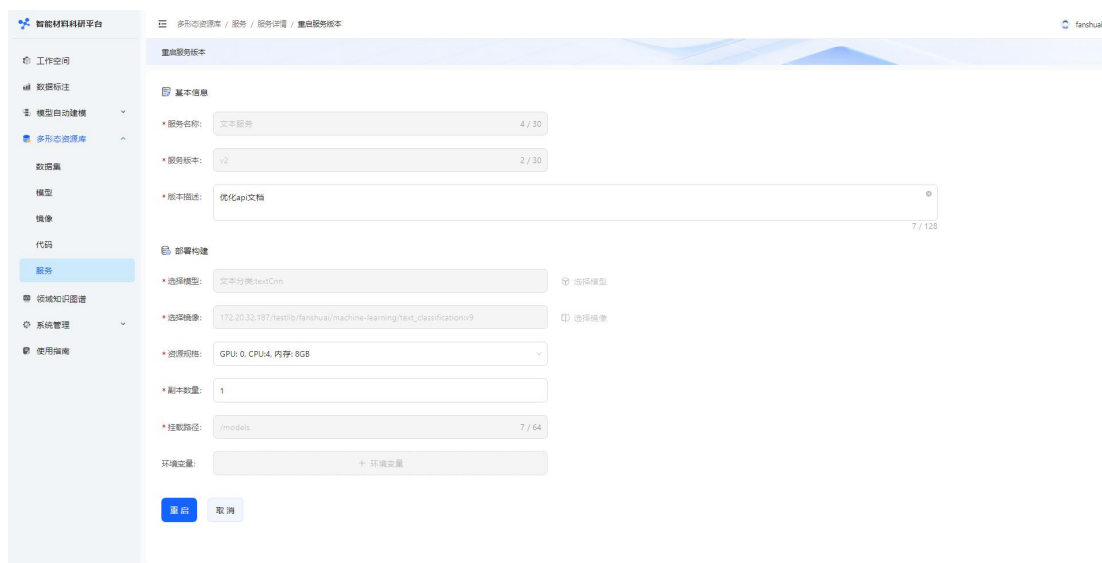


图 2-135 重启服务版本



图 2-136 停止服务版本

### 2.13.8 服务详情

查看服务版本基本信息、模型推理预测、调用指南和服务日志。

#### 步骤:

1. 服务列表界面点击操作【查看详情】按钮跳转到【服务详情】界面。
2. 服务版本列表点击操作【详情】按钮跳转到【服务详情】界面，可以查看到模型服务基本信息，包括模型的名称、模型的版本名称、模型的资源配置、模型的副本数和模型的 api 地址等，如图 2-137 所示。
3. 点击【预测】，用户可以通过 API 接口输入数据并获取预测结果，适用于图像分类、文本分析等多种应用场景，如图 2-138 所示。
4. 点击【调用指南】，为用户提供详细的服务调用说明，包括 API 接口文档、请求示例和参数说明，便于快速集成和使用服务，如图 2-139 所示。  
【预测】和【调用指南】页面可在基础环境镜像上，由用户使用 gradio 框架自定义开发，开发完成打包成新增服务版本的部署构建镜像，开发文档见附录。
5. 点击【服务日志】实时查看服务的运行日志，帮助用户监控服务状态并快速定位问题，如图 2-140 所示。



图 2-137 文本服务版本详情



图 2-138 文本服务版本详情预测

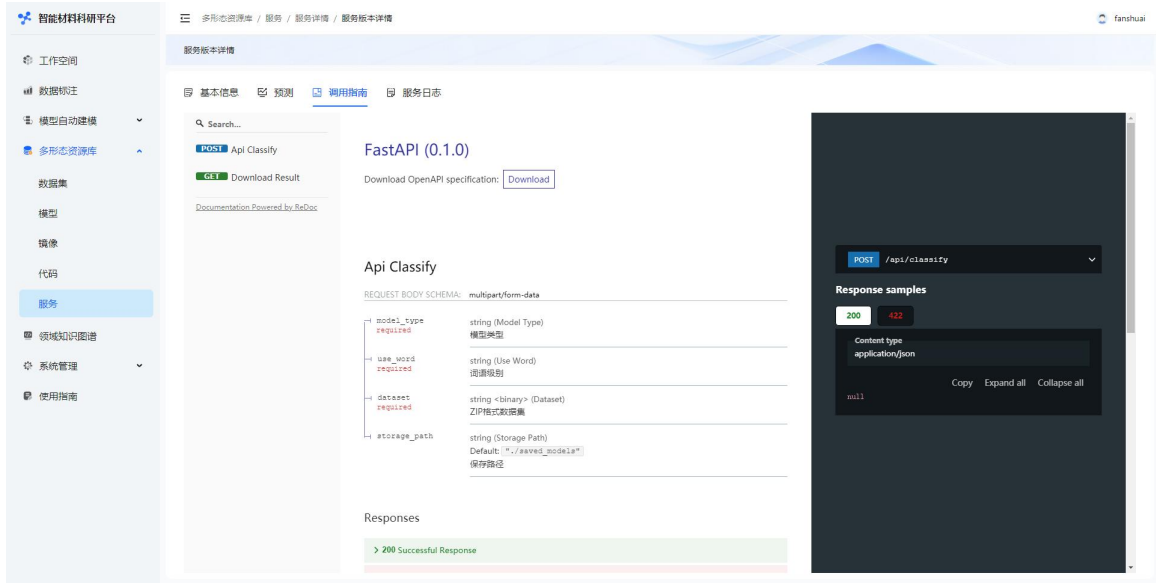


图 2-139 文本服务版本详情调用指南

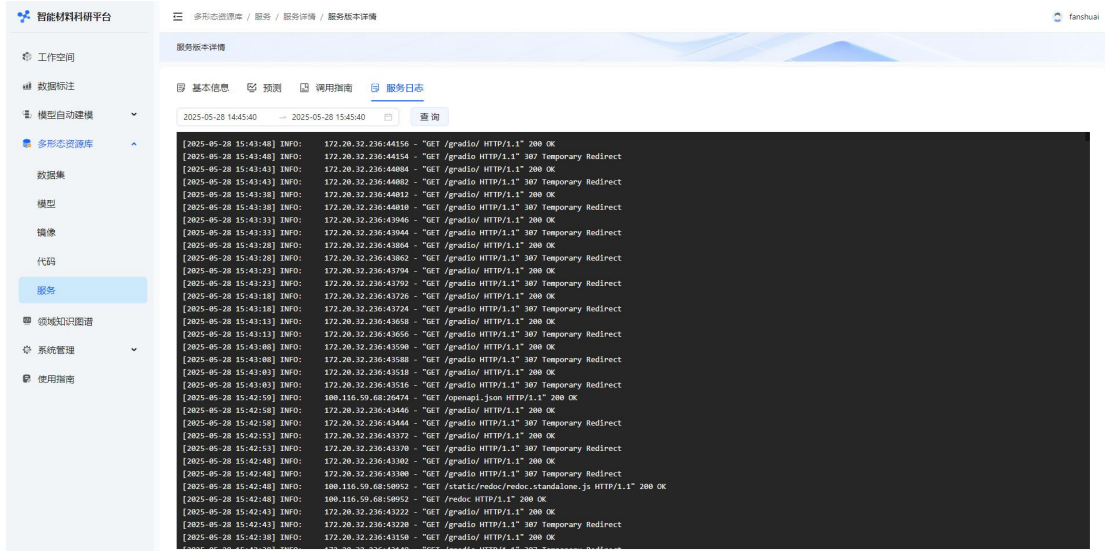


图 2-140 文本服务版本详情服务日志

### 2.13.9 服务版本对比

对比两个不同版本服务之间的差异，直观展示配置参数的变化，帮助用户了解版本的差异。

#### 步骤：

1. 服务列表界面点击操作【查看详情】按钮跳转到【服务详情】界面。
2. 服务版本列表勾选两个版本，点击【版本对比】按钮弹出【服务版本对比】弹窗，展示配置参数的差异，如图 2-141 所示。

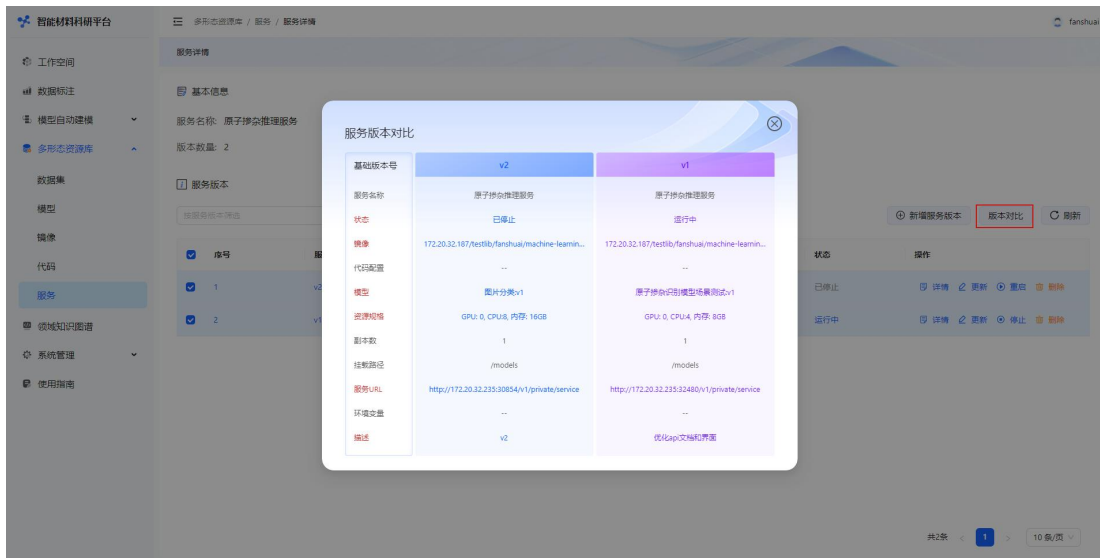


图 2-141 服务版本对比

### 2.13.10 删除服务版本

删除模型服务版本，系统会确保删除操作的安全性，避免误删服务版本。

步骤：

1. 服务列表界面点击操作【查看详情】按钮跳转到【服务详情】界面。
2. 点击服务版本列表操作【删除】-【确认】按钮成功删除服务版本，如图

2-142 所示。

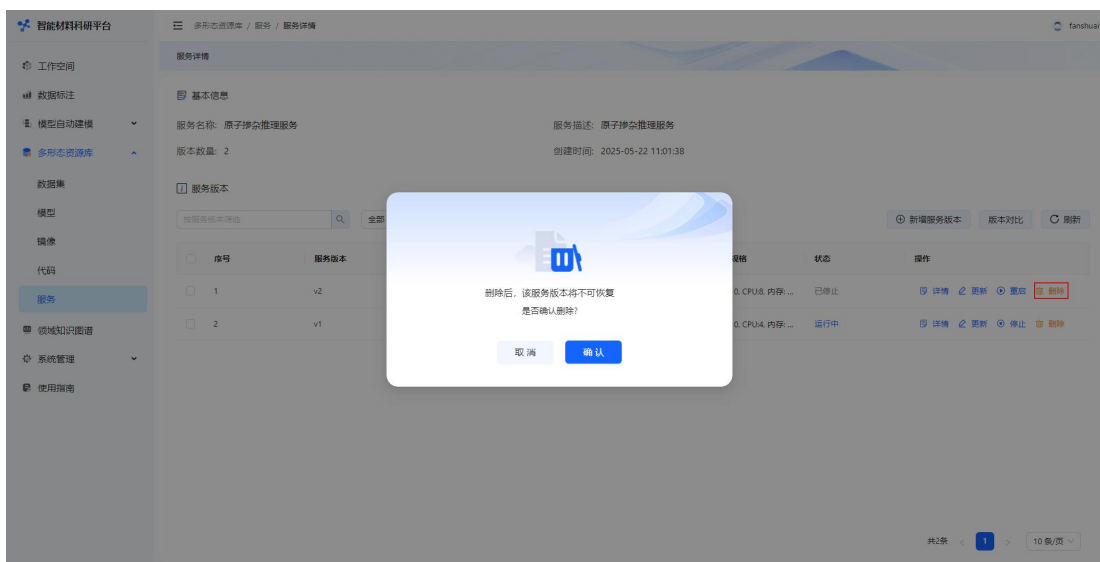


图 2-142 删除服务版本

### 2.13.11 删除服务

删除服务，系统会确保删除操作的安全性，若是服务中还存在多个版本，会提示该服务下还有版本，不能删除，避免误删服务。

## 步骤:

服务列表界面点击操作【删除】-【确认】按钮成功删除服务，如图 2-143 所示。

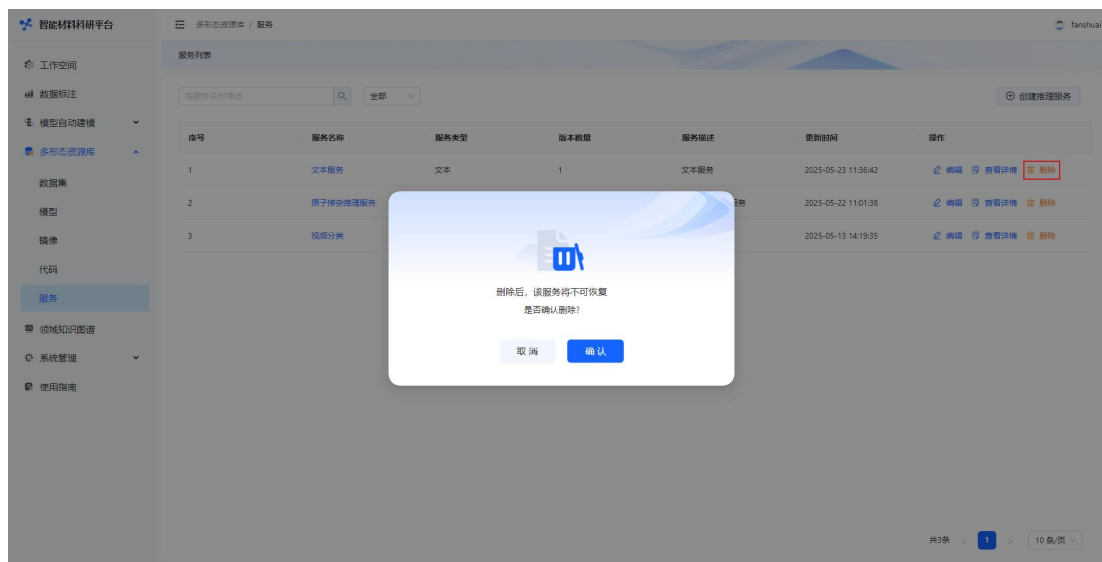


图 2-143 删除服务

## 2.14 领域知识图谱

本模块为用户提供了一套完整的知识图谱构建与管理工具，支持从零开始定义材料科学领域的核心概念（如材料成分、晶体结构、物理性能等）及其属性与关系，并通过动态更新机制确保本体的灵活性和时效性。

### 2.14.1 查看领域知识图谱模块

介绍用户如何进入领域知识图谱模块界面。

## 步骤:

点击菜单栏【领域知识图谱】菜单进入图空间列表界面，如图 2-144 所示。

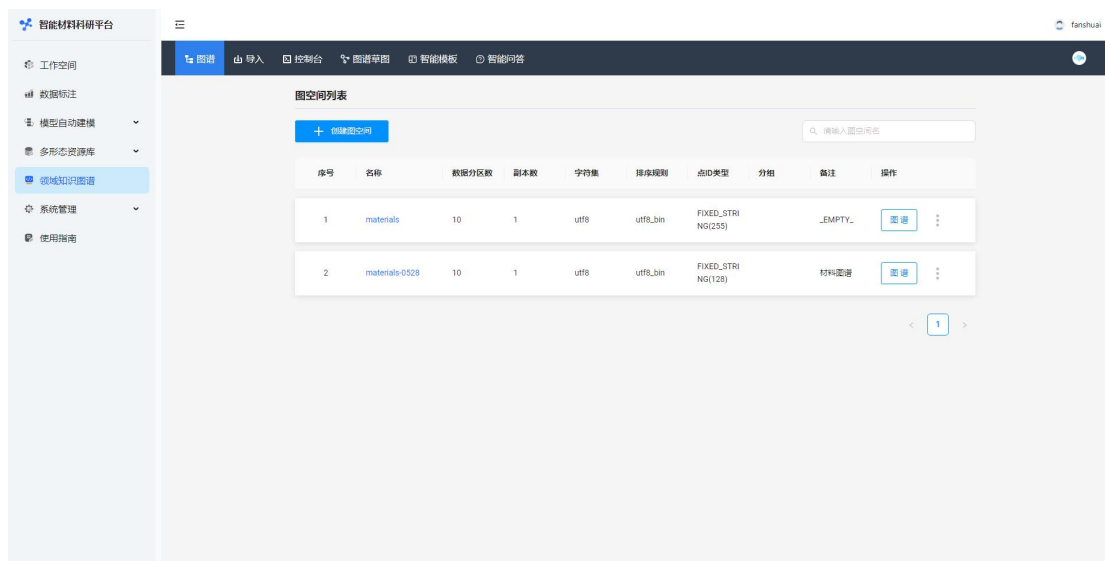


图 2-144 领域知识图谱模块界面

## 2.14.2 新增知识图谱图空间

构建材料科学领域的知识图谱图空间。

### 步骤:

1. 图空间列表界面点击【创建图空间】按钮跳转到【创建图空间】界面，如图 2-145 所示。
2. 输入名称和点 ID 类型，可选输入描述、数据分区数和副本数，点击【创建】按钮成功创建。
3. 点击图空间【名称】或操作列【图谱】按钮进入图空间，如图 2-146 所示。
4. 点击【标签】界面【创建】按钮跳转到【创建标签】界面，如图 2-147 所示，输入相关配置，点击【创建】按钮成功创建标签。
5. 点击【边类型】-【创建】按钮跳转到【创建边类型】界面，如图 2-148 所示，输入相关配置，点击【创建】按钮成功创建边类型。

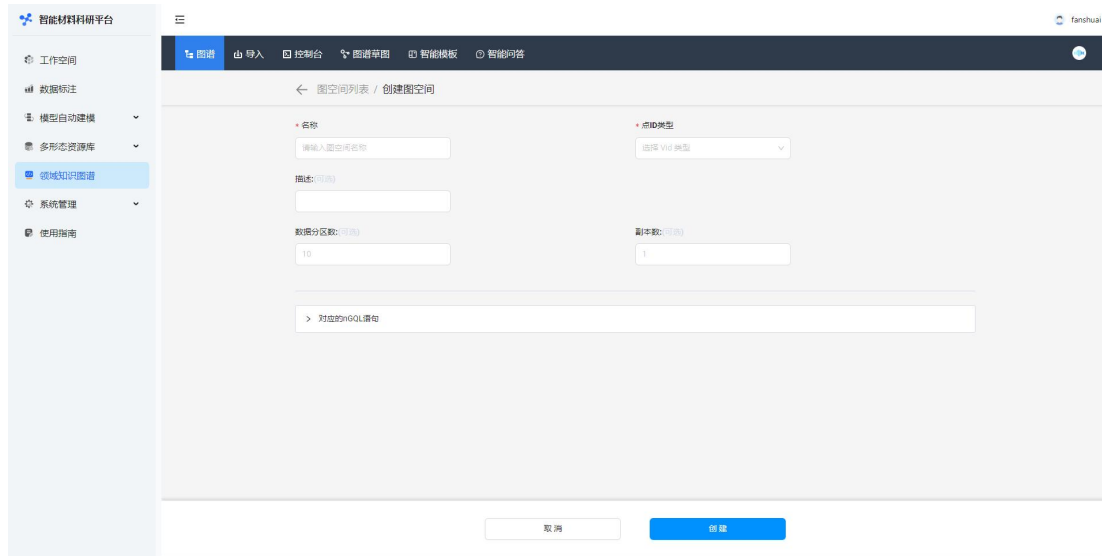


图 2-145 创建图空间

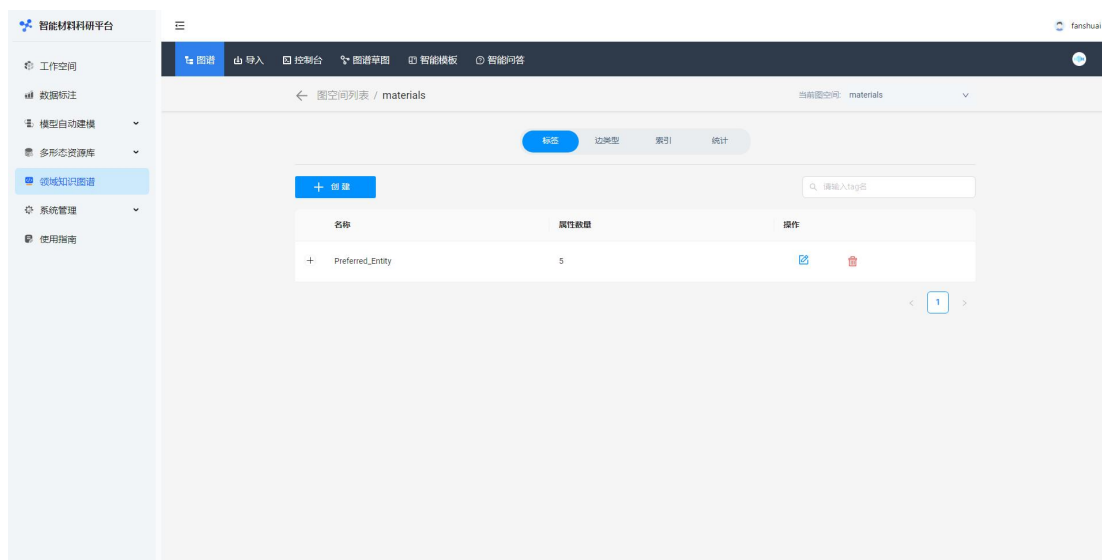


图 2-146 图空间

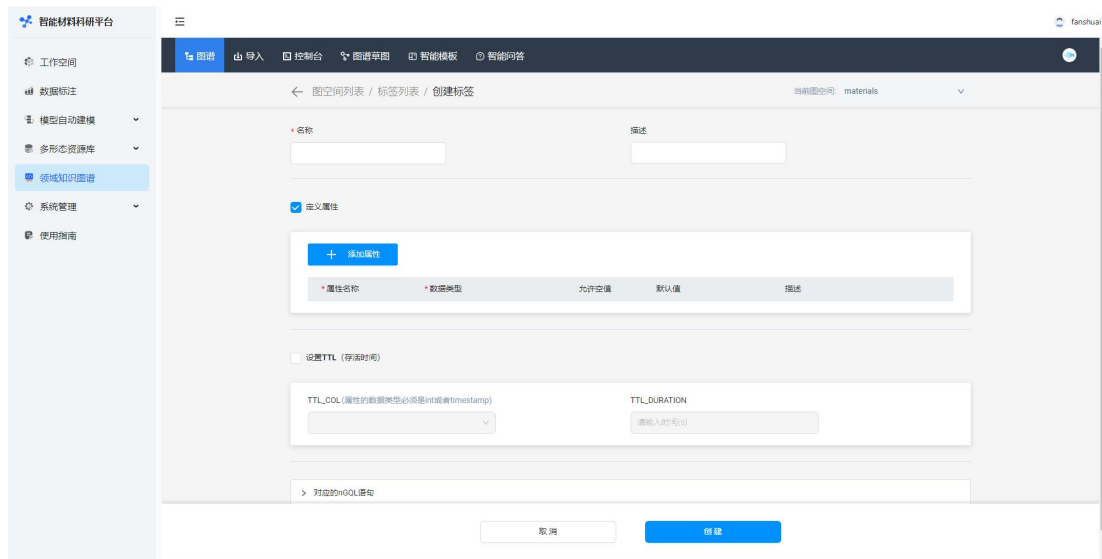


图 2-147 创建标签

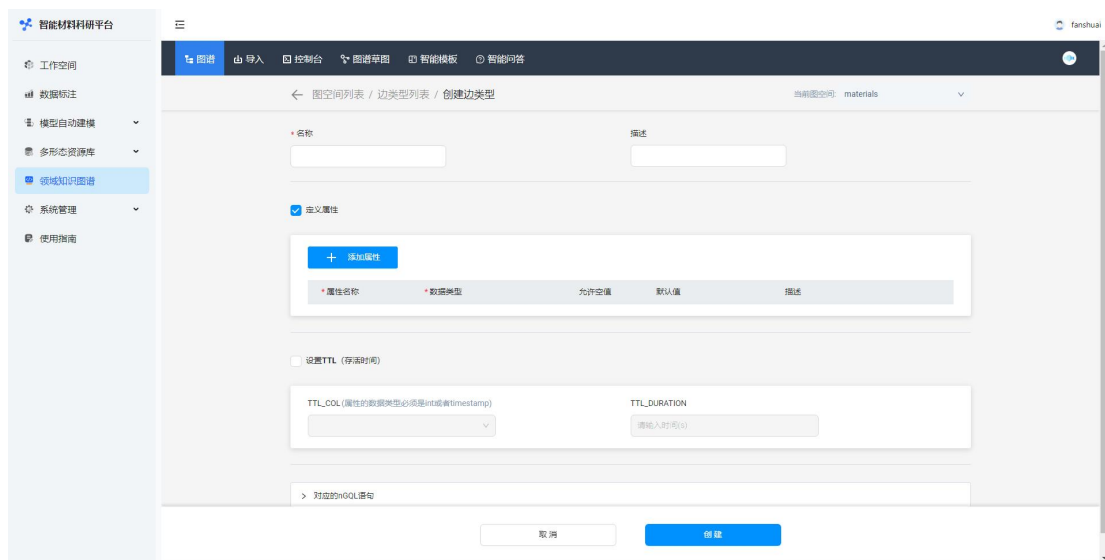


图 2-148 创建边类型

### 2.14.3 导入知识图谱数据

从外部数据源导入表格知识图谱数据，丰富现有知识图谱，支持全量增量更新。

#### 步骤：

1. 图空间列表界面点击【导入】按钮跳转到【导入数据】界面，如图 2-149 所示。
2. 导入数据需要先添加数据源，点击【数据源管理】菜单跳转到【数据源管理】界面，如图 2-150 所示。点击【上传文件】按钮选择导入的数据文件成功上传。
3. 点击【导入数据】菜单返回【导入数据】界面，点击【创建导入任务】按钮跳转到【创建导入任务】界面，选择导入的【图空间】，如图 2-151 所示。
4. 点击【关联标签】的【添加 Tag】-【选择 Tag】-【添加导入文件】-【选择绑定文件】按钮弹出【选择数据源文件】弹窗，如图 2-152 所示。数据源类型选择本地文件，点击标签需绑定文件，点击【添加】-【确认】按钮关联绑定文件成功。
5. 点击【选择 CSV 列】按钮选择 VID 列和点击【选择】按钮选择属性的对应列标。
6. 点击【关联边】的【添加 Edge Type】-【Edge 类型】-【添加导入文件】

- 【选择绑定文件】按钮弹出【选择数据源文件】弹窗。数据源类型选择本地文件，点击边需绑定文件，点击【添加】-【确认】按钮关联绑定文件成功。

7. 点击【选择 CSV 列】按钮选择起点 VID 列和终点 VID 列，点击【选择】按钮选择属性对应列标

8. 关联标签和关联边完成如图 2-153 所示，点击【导入】按钮弹出【导入任务确认】弹窗如图 2-154 所示，点击【确认】按钮返回【导入数据】任务列表界面，任务列表新增导入任务并开始读进度条，进度条 100%时完成导入知识图谱数据。

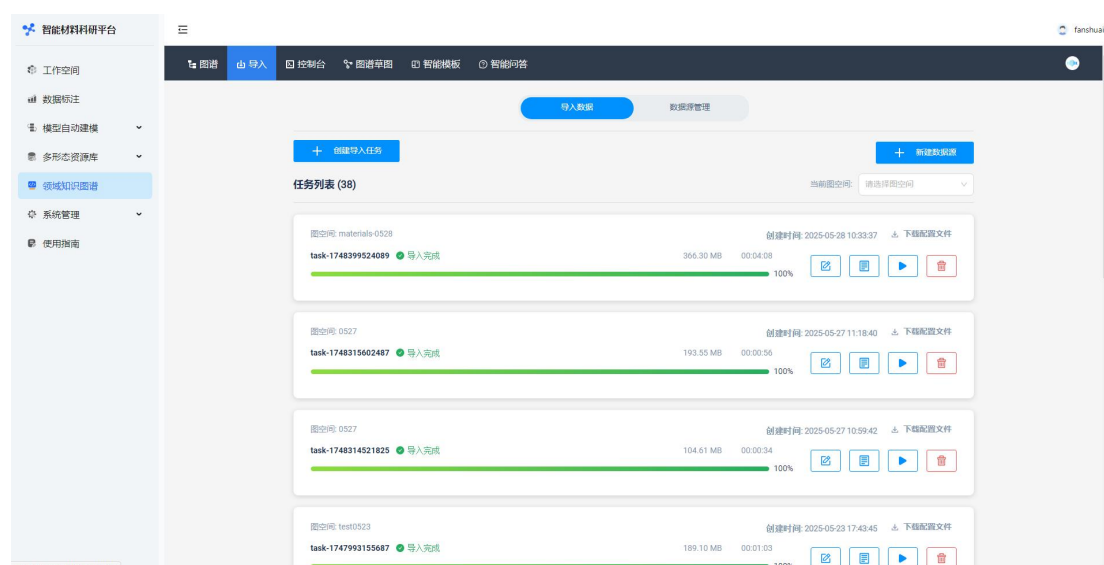


图 2-149 导入数据界面

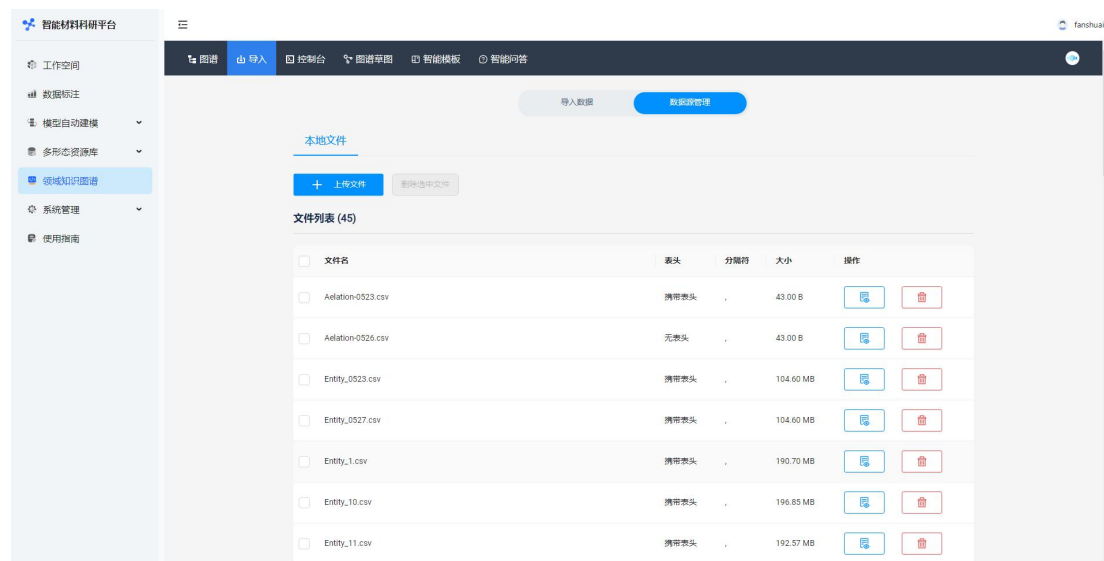


图 2-150 数据源管理界面

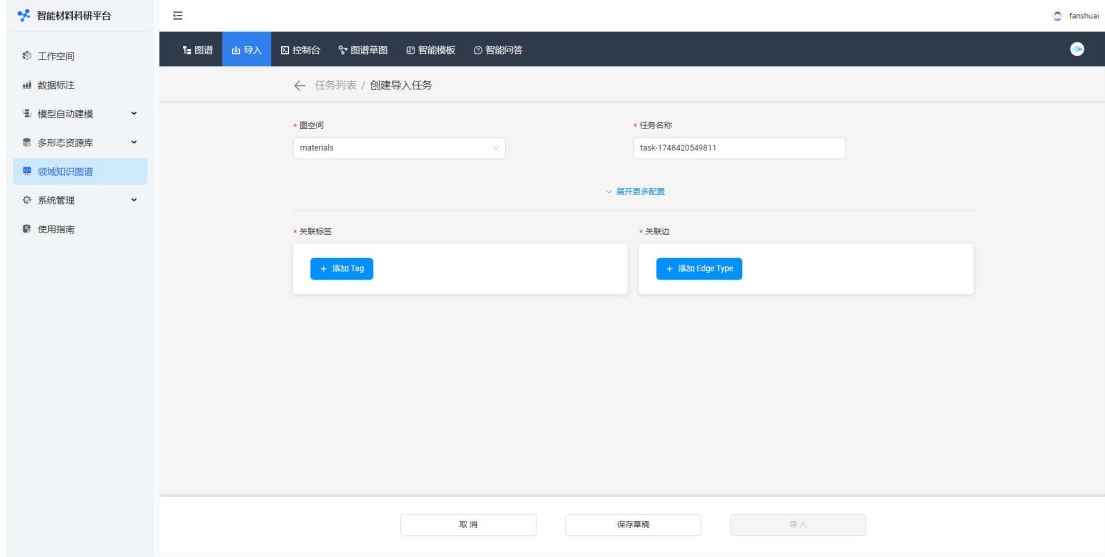


图 2-151 创建导入任务

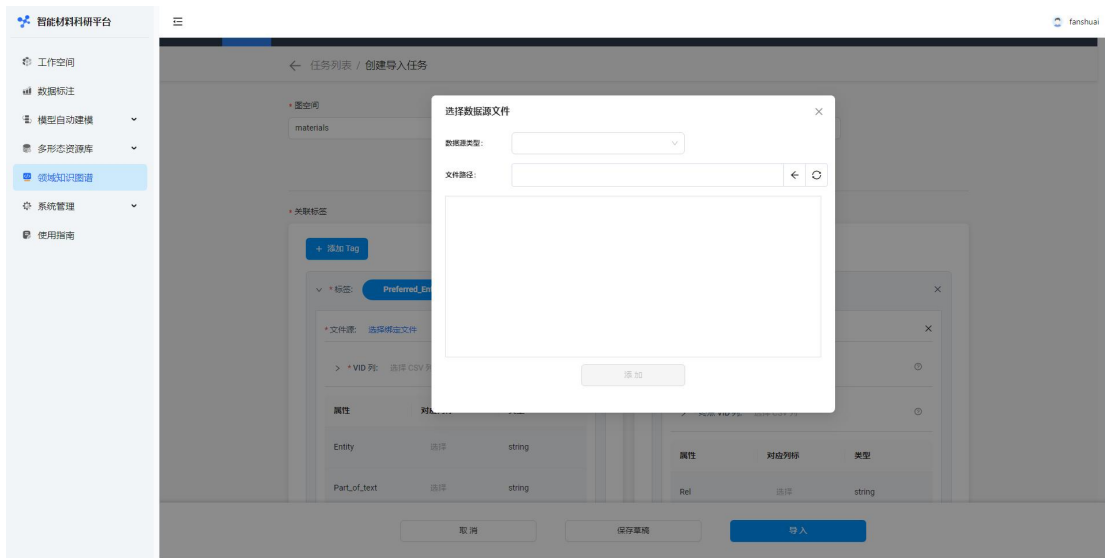


图 2-152 选择数据源文件

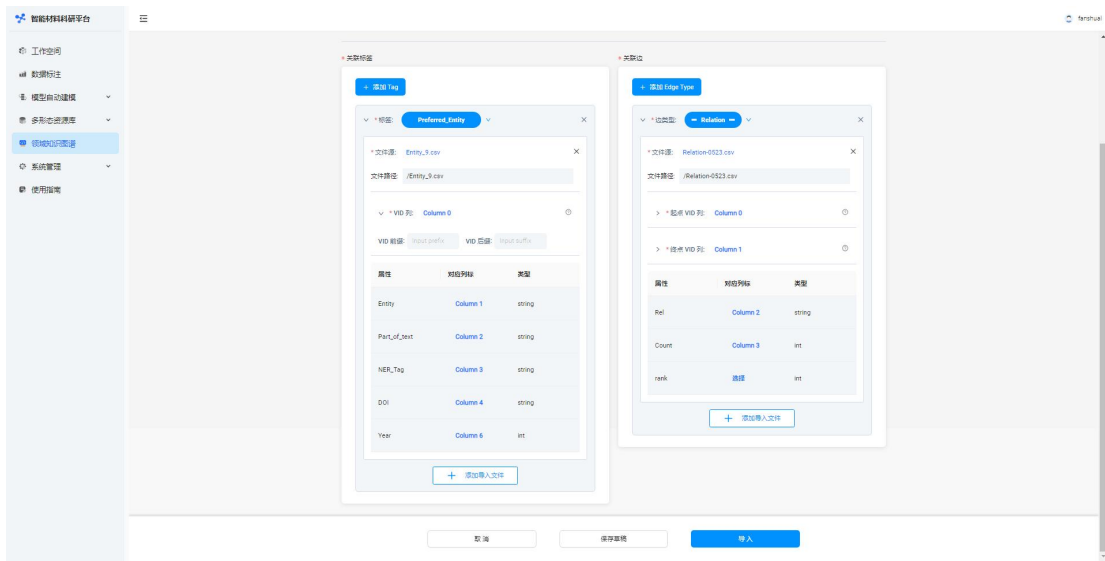


图 2-153 关联标签和关联边完成

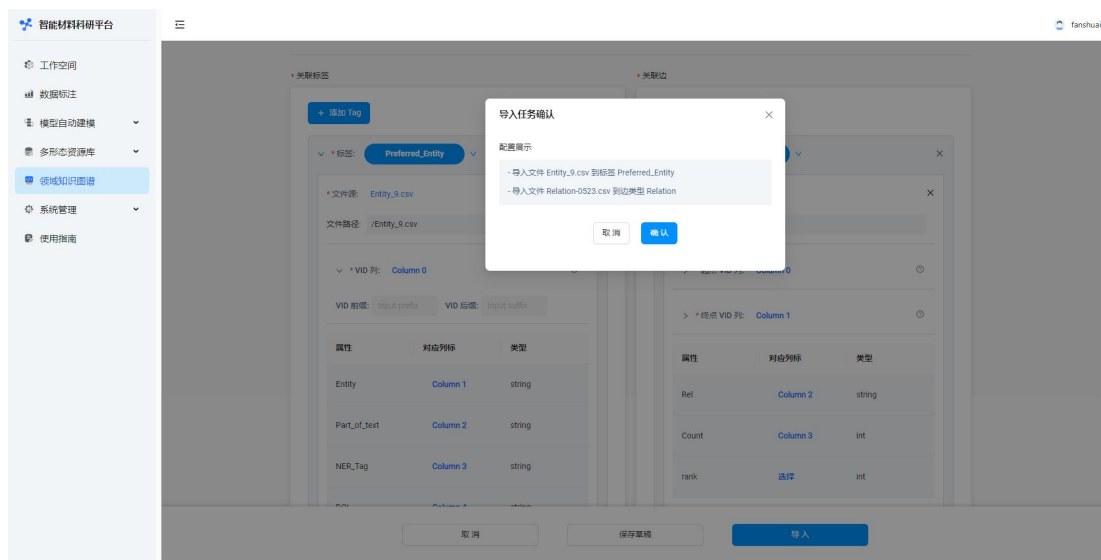



图 2-154 导入任务确认

### 2.14.4 删除知识图谱

选择不再需要的知识图谱进行删除操作，释放存储空间，同时系统会确保删除操作的安全性。

步骤：

图空间列表界面点击操作【】-【删除图空间】-【确认】按钮成功删除图空间，如图 2-155 所示。

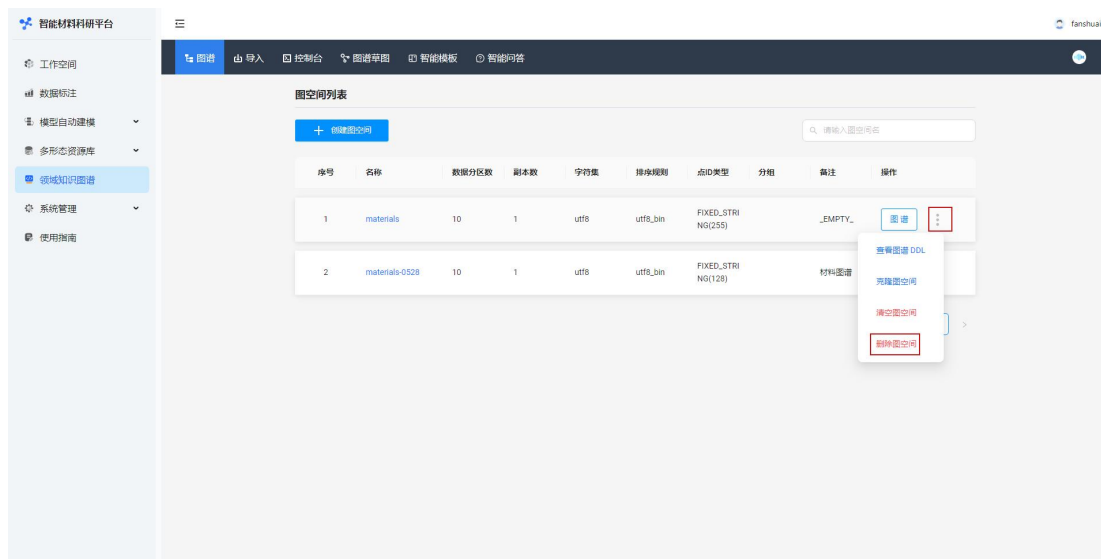


图 2-155 删除知识图谱

### 2.14.5 知识图谱控制台

控制台是用于管理和监控知识图谱数据库的工具，提供了图形化界面以及命

令行接口的支持，方便用户对数据库进行配置管理、性能优化、状态查看等一系列操作。

**步骤：**

图空间列表界面点击【控制台】菜单进入控制台界面，选择图空间，命令框输入命令，点击【运行】按钮将在选择图空间数据库执行对应命令，下方显示命令执行结果，如图 2-156 所示。

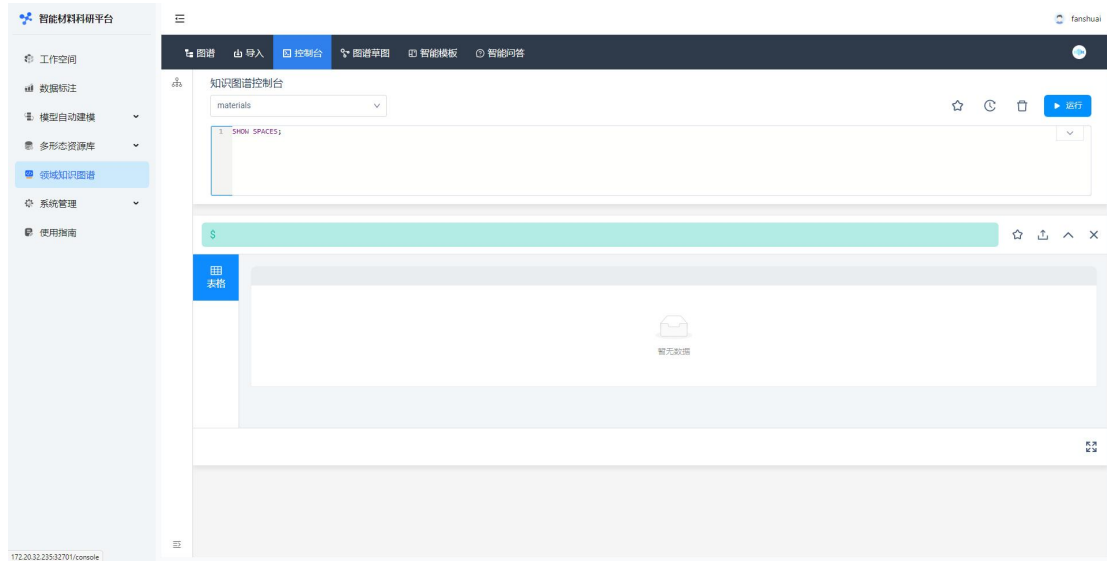


图 2-156 知识图谱控制台

### 2.14.6 智能模板配置

用户可以根据领域需求自定义问答模板，优化系统对特定类型问题的处理能力。

**步骤：**

图空间列表界面点击【智能模板】菜单进入智能模板配置界面，选择图空间，根据领域需求自定义编辑【提问模板】和【回答模板】，点击【更新】按钮配置成功，如图 2-157 所示。

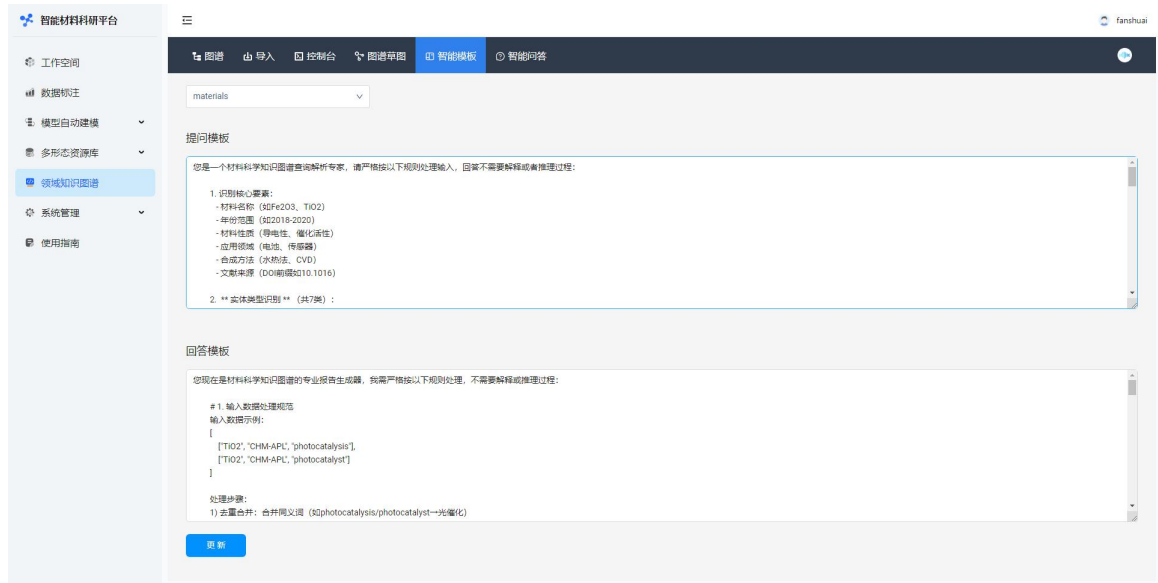


图 2-157 智能模板配置

### 2.14.7 知识问答推理

知识问答推理为用户提供基于知识图谱的智能化语义检索、问答和推理能力。用户可以通过自然语言输入查询，系统能够从材料领域知识图谱中精准定位相关实体、属性和关系，并生成结构化的答案。

**步骤：**

图空间列表界面点击【智能问答】菜单进入智能问答界面，点击【创建新对话】按钮创建新对话，问题输入框选择提问的图空间，输入问题后点击【提交】图标，系统将生成答案，如图 2-158 所示。

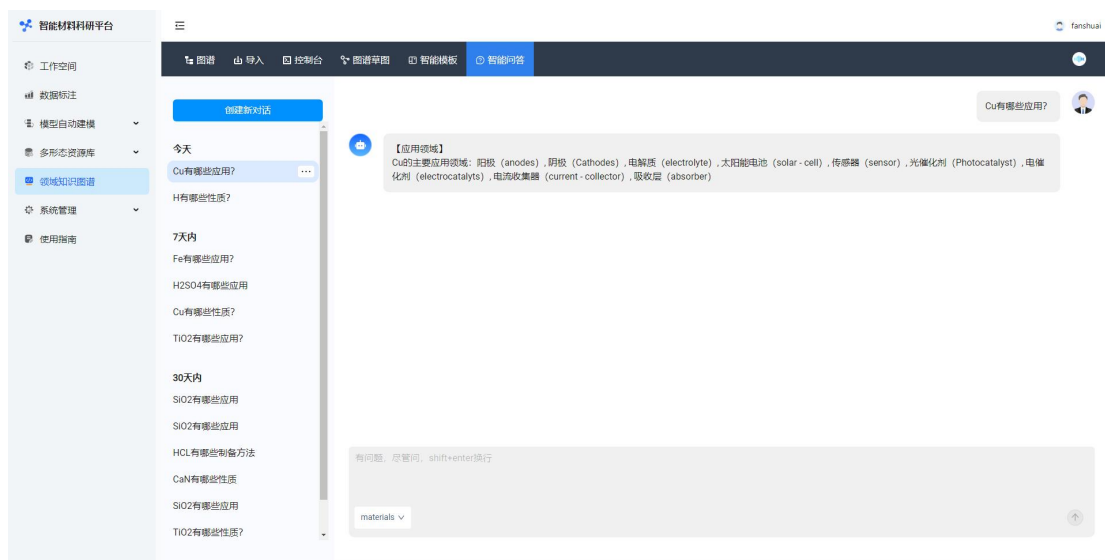


图 2-158 知识问答推理

# 附录

---

## gradio

---

### 简单入门

---

```
from fastapi import FastAPI
from fastapi.staticfiles import StaticFiles
import gradio as gr
import uvicorn

# 创建 FastAPI 应用
app = FastAPI()

# 文本处理函数
def process_text(text):
    """简单的文本处理函数"""
    return f"处理结果: {text.upper()}"

# 问候函数
def greet(name):
    """简单的问候函数"""
    return f"你好, {name}!"

# 创建 Gradio 界面
with gr.Blocks(title="Gradio 演示") as gradio_demo:
    with gr.Tab("文本处理"):
        gr.Markdown("## 文本处理工具")
        with gr.Row():
            text_input = gr.Textbox(label="输入文本")
            text_output = gr.Textbox(label="处理结果")
        text_button = gr.Button("处理文本")

        text_button.click(
            fn=process_text,
            inputs=text_input,
            outputs=text_output
        )

    with gr.Tab("问候"):
        gr.Markdown("## 问候生成器")
        with gr.Row():
            name_input = gr.Textbox(label="你的名字")
            greeting_output = gr.Textbox(label="问候语")
        greet_button = gr.Button("生成问候")

        greet_button.click(
            fn=greet,
            inputs=name_input,
            outputs=greeting_output
```

```

)

# 将 Gradio 应用挂载到 FastAPI
app.mount("/gradio", gradio_demo.app)

# FastAPI 路由
@app.get("/")
async def root():
    return {"message": "欢迎使用 FastAPI 和 Gradio 整合项目"}

@app.get("/v1/private/service")
async def api_hello(name: str = "World"):
    return {"message": f"Hello, {name}!"}

if __name__ == "__main__":
    # 启动 FastAPI 应用
    uvicorn.run(app, host="0.0.0.0", port=1888)

```

## 项目样例

完整代码链接如下：

<https://gitlink.org.cn/somunslotus/video-classification-service.git> slowfast分支

```

from fastapi.staticfiles import StaticFiles
from fastapi.openapi.docs import (
    get_redoc_html,
    get_swagger_ui_html,
    get_swagger_ui_oauth2_redirect_html,
)
app = FastAPI(docs_url=None, redoc_url=None)

static_dir = os.path.dirname(os.path.abspath(__file__))
app.mount("/static", StaticFiles(directory=f"{static_dir}/static"), name="static")

# 构建推理模型结构
model = create_slowfast(
    model_num_class=400,      # Kinetics-400数据集的类别数
    model_depth=50,          # ResNet-50作为骨干网络
    slowfast_fusion_conv_kernel_size=(7, 1, 1),
    slowfast_fusion_conv_stride=(4, 1, 1),
)

# 导入模型预设的权重
checkpoint = torch.load('SLOWFAST_8x8_R50.pyth', map_location='cpu')
model.load_state_dict(checkpoint['model_state'])
model.eval()

# 创建Gradio界面
# 编写gradio界面的标准格式
with gr.Blocks as demo:
    gr.Markdown("### 视频内容分类系统")
    # 水平分割
    with gr.Row():
        # 垂直分割，输入为video组件，包含一些对应的属性
        with gr.Column():

```

```

video_input = gr.Video(label="上传视频文件", sources=["upload"])
# 垂直分割, 输入为lable组件, 包含一些对应的属性
with gr.Column():
    output_label = gr.Label(label="分类结果TOP5", num_top_classes=5)
submit_btn = gr.Button("开始分析")
submit_btn.click(fn=video_classifier, inputs=video_input, outputs=output_label)

# 挂载的前端页面地址
app = gr.mount_gradio_app(app, demo, path="/gradio")

# 服务启动入口, 定义服务主程序所在的文件, ip和端口
if __name__ == "__main__":
    # 预设启动参数, 设置启动文件, ip和port
    uvicorn.run(
        "app:app",
        host="0.0.0.0",
        port=1888,
        reload=True, # 开发环境建议开启热重载
    )

```

运行效果如下

浏览器地址 : localhost:1888/gradio

#### 视频内容分类系统



注意：智能材料科研平台对外暴露的端口为1888，生成镜像时，会随机生成对外端口映射，且约定api的路径为/v1/private/service，gradio页面挂载地址为/gradio

以智能材料科研平台的视频分类为例：

服务版本详情

**基本信息** | 预测 | 调用指南 | 服务日志

服务名称: 视频分类

镜像: 172.20.32.187/testlib/fanshuai/machine-learning/video:v6

模型: 视频分类:v2

挂载路径: /models

文档地址: http://172.20.32.235:31693/redoc

创建时间: 2025-05-29 19:24:06

环境变量: --

版本名称: v6

状态: 运行中

资源规格: GPU: 0, CPU:4, 内存: 8GB

API URL: http://172.20.32.235:31693/v1/private/service

副本数量: 1

更新时间: 2025-05-30 08:43:00

描述: v6

# Interface 类

`gr.Interface` 类是 Gradio 中的高级抽象，只需指定输入类型和输出类型，即可快速为任何 Python 函数创建用户界面 (UI)。Interface 类有三个核心参数：

1. `fn`：需要包裹 UI 的函数。`fn` 参数非常灵活，可以是任何你想要包裹 UI 的 Python 函数，例如音乐生成器、税务计算器，或者是预训练机器学习模型的预测函数。函数的每个参数对应于一个输入组件，并且函数应返回单个值或值的元组，元组中的每个元素对应于一个输出组件。
2. `inputs`：输入的 Gradio 组件或组件列表。组件可以作为实例化对象传递，也可以通过其字符串快捷方式引用。组件的数量与函数参数的数量相匹配。
3. `outputs`：输出的 Gradio 组件或组件列表。组件的数量与函数返回值的数量相匹配

## 组件

Gradio 提供了超过 30 个内置组件，以及许多用户自定义的组件（详见 [gradio Components](#)）。这些组件涵盖了机器学习和数据科学中常见的数据类型，例如，`gr.Image` 组件用于处理输入或输出图像，`gr.Label` 组件用于显示分类标签和概率，`gr.Plot` 组件用于显示各种类型的图表等

1. 输入组件：如 `Textbox`（文本框）、`Number`（数字输入框）、`Image`（图像上传）、`Audio`（音频上传/播放）、`Checkbox`（复选框）、`Radio`（单选按钮）、`Dropdown`（下拉菜单）等。这些组件用于接收用户的输入。
2. 输出组件：如 `Label`（标签，用于显示文本）、`Image`（图像显示）、`Audio`（音频播放）、`Markdown`（Markdown 格式文本显示）、`Gallery`（图像画廊）等。这些组件用于展示处理结果或信息。
3. 其他组件：如 `Button`（按钮，用于触发事件）、`Slider`（滑动条，用于选择数值）、`State`（不可见组件，用于在后台存储变量）等

## Textbox

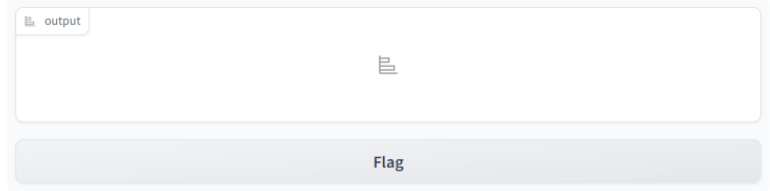
部分参数	描述
<code>value</code>	文本框的默认文本，可以是一个字符串或者一个在应用加载时调用的函数来设置初始值
<code>placeholder</code>	文本框的占位符提示文本
<code>label</code>	组件在界面中的名称
<code>max_lines</code>	文本框的最大行数，默认为 20
<code>type</code>	文本框的类型，可以是 <code>'text'</code> ， <code>'password'</code> ， <code>'email'</code> ，默认为 <code>'text'</code>
<code>visible</code>	是否可见，默认为 <code>True</code>

## Image

```
import gradio as gr

def image_classifier(inp):
    return {'cat': 0.3, 'dog': 0.7}
```

```
demo = gr.Interface(fn=image_classifier, inputs="image", outputs="label")
demo.launch()
```



部分参数	描述
value	用于设置组件的默认值，可以是PIL图像、numpy数组、文件路径或URL
shape	用于裁剪和调整图像的形状，可以指定宽度和高度
height	显示图像的高度（像素）
width	显示图像的宽度（像素）
image_mode	图像模式，可以是“RGB”（彩色）或“L”（黑白）
source	图像的来源，可以是“upload”（上传）、“webcam”（摄像头）或“canvas”（画布）

## gradio block类

### 示例1

Gradio 提供了一种低级方法，用于使用 gr.Blocks 类设计具有更可自定义布局和数据流的 Web 应用程序。Blocks 支持诸如控制组件在页面上的显示位置、处理多个数据流和更复杂的交互（例如，输出可以用作其他函数的输入），以及基于用户交互更新组件的属性/可见性

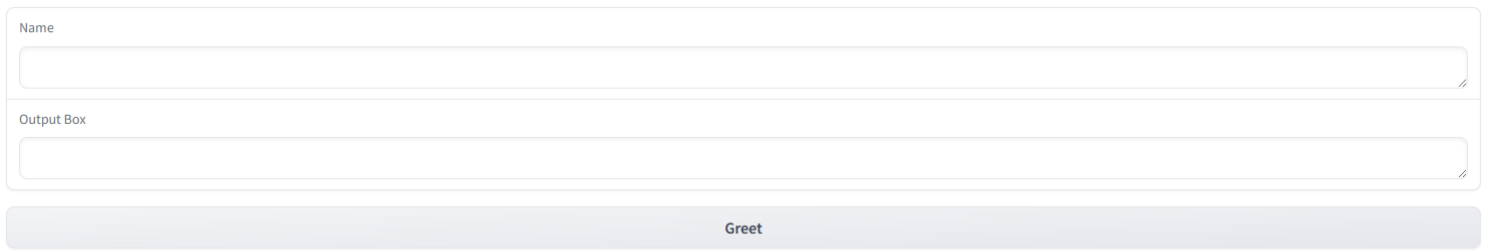
```
import gradio as gr

def greet(name):
    return "Hello " + name + "!"

with gr.Blocks() as demo:
    name = gr.Textbox(label="Name")
    output = gr.Textbox(label="Output Box")
    greet_btn = gr.Button("Greet")
    greet_btn.click(fn=greet, inputs=name, outputs=output, api_name="greet")

if __name__ == "__main__":
    demo.launch(server_port=7860)
```

代码启动后，浏览器访问<http://127.0.0.1:7860/>，即可看到如下界面



The screenshot shows a web interface with two input fields and a button. The top field is labeled 'Name' and is empty. Below it is a field labeled 'Output Box', also empty. At the bottom is a button labeled 'Greet'.

代码解释：

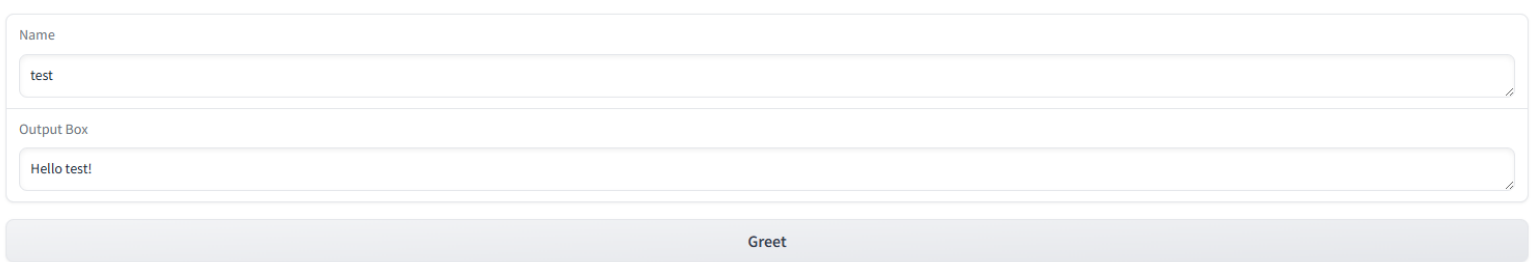
`with gr.Blocks() as demo`：定义一个block类

`output = gr.Textbox(label="Output Box")`：定义一个文本框，名为Output Box

`greet_btn = gr.Button("Greet")`：定义一个按钮

`greet_btn.click(fn=greet, inputs=name, outputs=output, api_name="greet")`：定义一个按钮的点击事件，函数内inputs为name的文本框，输出为output的文本框，且关联的处理逻辑为函数greet

汇总后即可得：在Name的输入框中输入任意文本，则在点击按钮后，文本传递到greet函数处理，处理结果显示在Output Box文本框内



The screenshot shows the same web interface as before, but now the 'Name' field contains the text 'test' and the 'Output Box' contains the text 'Hello test!'. The 'Greet' button is still visible at the bottom.

## 示例2：布局

### 行

在 `with gr.Row` 子句中的元素都将水平显示。例如，要并排显示两个按钮：

```
with gr.Blocks() as demo:
    with gr.Row():
        btn1 = gr.Button("Button 1")
        btn2 = gr.Button("Button 2")
```

可以设置行中每个元素都具有相同的高度。使用 `equal_height` 参数配置此项

```
with gr.Blocks() as demo:
    with gr.Row(equal_height=True):
        textbox = gr.Textbox()
        btn2 = gr.Button("Button 2")
```

可以通过每个组件中都存在的 `scale` 和 `min_width` 参数的组合来控制行中元素的宽度

`scale` 是一个整数，用于定义元素在行中占用空间的方式。如果 `scale` 设置为 `0`，则元素不会扩展以占用空间。如果 `scale` 设置为 `1` 或更大，则元素将扩展。行中的多个元素将根据其 `scale` 成比例地扩展。在下面，`btn2` 将扩展为 `btn1` 的两倍，而 `btn0` 将不会扩展

```
with gr.Blocks() as demo:
    with gr.Row():
        btn0 = gr.Button("Button 0", scale=0)
        btn1 = gr.Button("Button 1", scale=1)
        btn2 = gr.Button("Button 2", scale=2)
```

`min_width` 将设置元素将占用的最小宽度。如果没有足够的空间来满足所有 `min_width` 值，则行将换行

## 列和嵌套

Column 中的组件将彼此垂直放置。由于垂直布局无论如何都是 Blocks 应用的默认布局，因此为了使其有用，Column 通常嵌套在 Row 中

```
import gradio as gr

with gr.Blocks() as demo:
    with gr.Row():
        text1 = gr.Textbox(label="t1")
        slider2 = gr.Slider(label="s2")
        drop3 = gr.Dropdown(["a", "b", "c"], label="d3")
    with gr.Row():
        with gr.Column(scale=1, min_width=300):
            text1 = gr.Textbox(label="prompt 1")
            text2 = gr.Textbox(label="prompt 2")
            inbtw = gr.Button("Between")
            text4 = gr.Textbox(label="prompt 1")
            text5 = gr.Textbox(label="prompt 2")
        with gr.Column(scale=2, min_width=300):
            img1 = gr.Image("images/cheetah.jpg")
            btn = gr.Button("Go")

demo.launch()
```

t1

s2

d3


prompt 1

prompt 2

Between

prompt 1

prompt 2



Go

第一个列有两个垂直排列的文本框。第二个列有一个图像和一个按钮垂直排列。请注意，两列的相对宽度是如何通过 `scale` 参数设置的。`scale` 值是两倍的列占据两倍的宽度

## 选项卡和手风琴

使用 `with gr.Tab('tab_name'):` 子句创建选项卡。在 `with gr.Tab('tab_name'):` 上下文中创建的任何组件都会显示在该选项卡中。连续的选项卡子句组合在一起，以便一次只能选择一个选项卡，并且仅显示该选项卡上下文中的组件

```
import numpy as np
import gradio as gr

def flip_text(x):
    return x[::-1]

def flip_image(x):
    return np.fliplr(x)

with gr.Blocks() as demo:
    gr.Markdown("Flip text or image files using this demo.")
    with gr.Tab("Flip Text"):
        text_input = gr.Textbox()
        text_output = gr.Textbox()
        text_button = gr.Button("Flip")
    with gr.Tab("Flip Image"):
        with gr.Row():
            image_input = gr.Image()
            image_output = gr.Image()
        image_button = gr.Button("Flip")

    with gr.Accordion("Open for More!", open=False):
        gr.Markdown("Look at me...")
        temp_slider = gr.Slider(
```

```

0, 1,
value=0.1,
step=0.1,
interactive=True,
label="Slide me",
)

text_button.click(flip_text, inputs=text_input, outputs=text_output)
image_button.click(flip_image, inputs=image_input, outputs=image_output)

demo.launch()

```

Flip text or image files using this demo.

## 可见性

组件和布局元素都具有 `visible` 参数，可以初始设置和更新。在 `Column` 上设置 `gr.Column(visible=...)` 可用于显示或隐藏一组组件

```

import gradio as gr

with gr.Blocks() as demo:
    name_box = gr.Textbox(label="Name")
    age_box = gr.Number(label="Age", minimum=0, maximum=100)
    symptoms_box = gr.CheckboxGroup(["Cough", "Fever", "Runny Nose"])
    submit_btn = gr.Button("Submit")

    with gr.Column(visible=False) as output_col:
        diagnosis_box = gr.Textbox(label="Diagnosis")
        patient_summary_box = gr.Textbox(label="Patient Summary")

    def submit(name, age, symptoms):
        return {
            submit_btn: gr.Button(visible=False),
            output_col: gr.Column(visible=True),
            diagnosis_box: "covid" if "Cough" in symptoms else "flu",
            patient_summary_box: f"{name}, {age} y/o",
        }

    submit_btn.click(
        submit,
        [name_box, age_box, symptoms_box],
        [submit_btn, diagnosis_box, patient_summary_box, output_col],
    )

```

```
)  
demo.launch()
```

## 流式输出

有时候，我们希望展示一系列连续的输出，而不是一次性显示单个结果。这种功能在某些场景下非常有用。

比如:图像生成模型：可以展示每一步生成的中间图像，直到最终图像完成。聊天机器人：可以一个词一个词地流式输出回复，而不是等待整个回答生成完毕才显示。

在Gradio中，可以使用Python的生成器函数(generator)来实现流式输出。生成器函数的特点是：使用yield语句而不是return语句；yield语句通常会放在循环中，以产生一系列的值。

例如，下面的生成器函数会依次生成从0到x-1的数字：

```
def my_generator(x):  
    for i in range(x):  
        yield i
```

在Gradio中使用生成器的方式与使用普通函数相同，下面是一个图像生成模型的例子

```
import gradio as gr  
import numpy as np  
import time  
  
def fake_diffusion(steps):  
    rng = np.random.default_rng()  
    for i in range(steps):  
        time.sleep(1) # 创建人工停顿，便于观察  
        image = rng.random(size=(600, 600, 3))  
        yield image  
    image = np.ones((1000,1000,3), np.uint8)  
    image[:] = [255, 124, 0]  
    yield image  
  
demo = gr.Interface(fake_diffusion,  
                    inputs=gr.Slider(1, 10, 3, step=1),  
                    outputs="image")  
  
demo.launch()
```

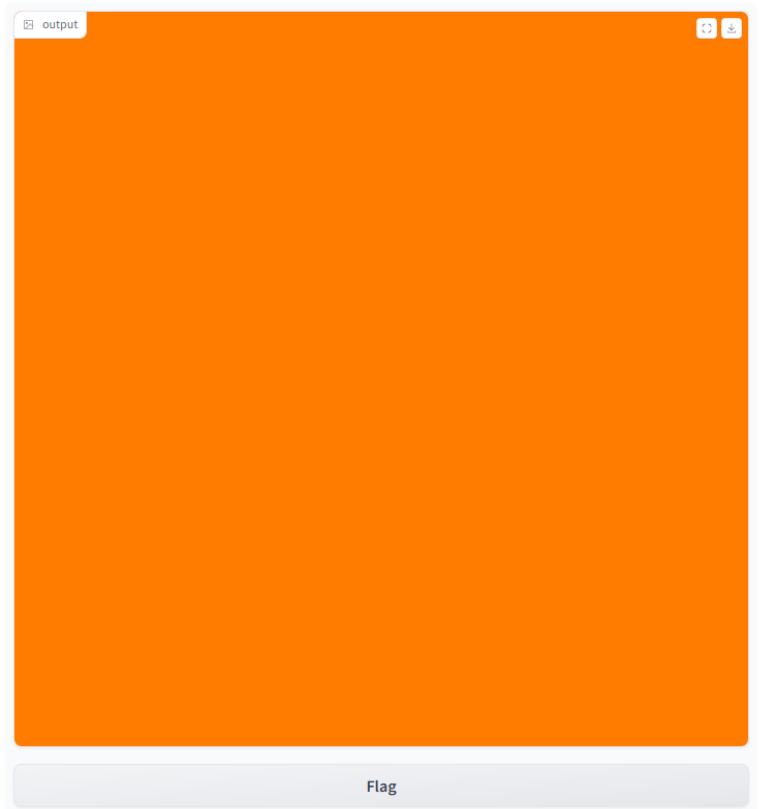
上述代码中的模型会生成几步随机噪声图像，最后输出一个橙色的图像。界面会有一个滑动条，用于控制步数。

- fake\_diffusion函数：
  - 这是一个生成器函数，接受步数作为参数
  - 循环指定的步数，每步生成一个随机噪声图像并yield
  - 最后生成一个橙色图像并yield
- 创建Gradio界面：

- 使用gr.Interface
- 输入是一个范围为1-10的滑动条
- 输出是图像

steps

Clear Submit



# Fastapi

## 样例，了解FastAPI程序结构

```
from fastapi import FastAPI # 导入FastAPI框架

app = FastAPI() # 创建一个app实例

@app.get("/") # 编写一个 路径操作装饰器
def root(): # 操作函数，位于路径操作装饰器下方
    return {"message": "Hello World"}
```

1. 使用 `uvicorn main:app --reload` 启动
2. 访问 <http://127.0.0.1:8000/>
3. 获取返回 `{"message": "Hello World"}`

- `main`: `main.py` 文件(也可理解为Python模块).
- `app`: `main.py` 中 `app = FastAPI()` 语句创建的app对象.
- `--reload`: 在代码改变后重启服务器，只能在开发的时候使用

- 你可以将get操作方法更改成 `@app.post()`、`@app.put()`、`@app.delete()` 等方法
- 你可以更改相应的路径 ("/") 为自己想要的，例如我更改为 ("/hello\_word/")

## 路径操作装饰器中的路径参数

---

### 路径操作装饰器中的路径参数

#### 声明路径参数

```
from fastapi import FastAPI

app = FastAPI()

@app.get("/items/{item_id}")
def read_item(item_id):
    return {"item_id": item_id}
```

上述代码运行之后，路径参数 `item_id` 的值会作为 `read_item` 函数参数 `item_id` 的值。

因此，如果你运行上述示例，然后跳转到 <http://127.0.0.1:8000/items/foo>，你将会看见这样的回应: `{"item_id": "foo"}`

#### 声明路径参数的类型

将参数 `item_id` 的类型定义为 `int` 类型

```
from fastapi import FastAPI

app = FastAPI()

@app.get("/items/{item_id}")
async def read_item(item_id: int):
    return {"item_id": item_id}
```

将参数 `item_name` 的类型定义为 `str` 类型

```
from fastapi import FastAPI

app = FastAPI()

@app.get("/items/{item_name}")
def read_item(item_name: str):
    return {"item_id": item_name}
```

## gradio整合fastapi

---

### 示例1

```

from fastapi import FastAPI
import gradio as gr

# 初始化 FastAPI 应用
app = FastAPI(title="Demo Service")

# ----- 共享业务逻辑 -----
def process_text(input_str: str) -> str:
    """核心处理逻辑：文本转大写 + 添加表情"""
    result = input_str.upper()
    return f"🌈 {result} 🌈"

# ----- FastAPI API 接口 -----
# 路径中接收一个文本参数
@app.get("/api/uppercase")
async def api_uppercase(text: str):
    """
    REST API 接口
    测试方法：浏览器访问 http://localhost:8000/api/uppercase?text=hello
    或使用 curl：
    curl "http://localhost:8000/api/uppercase?text=hello"
    """
    return {"original": text, "processed": process_text(text)}

# ----- Gradio 交互界面 -----
def gradio_callback(text):
    return process_text(text)

# 使用Interface组建前端页面
gradio_ui = gr.Interface(
    # 定义方法事件
    fn=gradio_callback,
    # 输入
    inputs=gr.Textbox(label="输入文本", placeholder="在这里输入文字..."),
    # 输出
    outputs=gr.Textbox(label="处理结果"),
    title="文本处理演示器",
    examples=[
        ["hello world"],
        ["fastapi is awesome"],
        ["gradio makes ui easy"]
    ],
    allow_flagging="never"
)

# ----- gradio和fastapi挂载整合 -----
# 定义路径后缀为/ui
app = gr.mount_gradio_app(app, gradio_ui, path="/ui")

# ----- 运行测试 -----
if __name__ == "__main__":
    import uvicorn
    uvicorn.run(app, host="localhost", port=8000)

```

浏览器输入路径：<http://localhost:8000/ui/>，得到如下界面

# 文本处理演示器

输入文本

在这里输入文字...

Clear

Submit

处理结果

☰ Examples

hello world

fastapi is awesome

gradio makes ui easy