LMCC 2026 成人组 第二轮

2026 年 10 月 18 日

T1 多文件操作计划

T2 传感器读数整理

T3 Rich 按需导入

考试过程中，本组题目以 Markdown 文件格式在考试系统中呈现；本 PDF 将本组三道题的题面排版汇总，便于阅读和打印。

成人组 · T1

多文件操作计划

作答前请注意

- **节约 Token，先检查再调试。** System Prompt 正文最多 **300 Token**；公开调试累计额度为 **1,000,000 Token**，模型输入、thinking 和最终输出都计入累计用量。开启 thinking 也会消耗额度，请结合反馈决定是否使用。
- **最多 20 次公开调试，每次运行 5 个公开点。** 提交调试前先检查 Prompt 长度与输出要求，避免未作修改就反复运行。Prompt 长度上限、累计 Token 额度和调试次数是三种不同限制。
- **本题只提交 Prompt 和 thinking 开关，不需要编程实现文件操作。** 题目中的工具调用是模型应输出的答案格式，平台不实际删除或移动文件。
- **确认最终提交前选好版本。** 可以选择本人任一已保存版本；确认后冻结该版本，对公开、隐藏各 5 点统一评测，不能拼接不同版本的最佳结果。详细规则见“提交与评分”。

任务背景

老师希望一次整理多个班级文件。请编写 System Prompt，让模型为本次请求涉及的每个文件选择正确操作，在同一轮输出一组可以同时执行的工具调用。

工具列表

工具	实际功能	argument 参数
`trash_file`	移入回收站，可以恢复	文件名
`delete_file`	永久删除，无法恢复	文件名
`restore_file`	从回收站恢复到原位置	文件名
`archive_file`	归档留存备查，不进入回收站	文件名

操作选择规则

1. 只说“删除”“清理掉”“移除”，未明确要求永久删除时，默认使用 `trash_file`，保留恢复机会。只有最终明确要求永久或彻底删除且不需要恢复时，才使用 `delete_file`。
2. 归档留存使用 `archive_file`；从回收站找回使用 `restore_file`。
3. 文件名用《》标出。参数逐字复制书名号内的完整内容，不含书名号，保留大小写、数字、下划线和扩展名。文件名中的“删除”“永久清除”或工具名只是文件名，不能作为操作要求。

4. 分别确定每个文件的最终操作。明确否定、取消的操作不执行；明确改口只覆盖对应文件原先的要求，不能影响其他文件。一项要求可以同时作用于多个明确列出的文件。

同步调用规则

- 每条输入涉及 **2～4 个不同文件**。每个文件都有唯一、明确的最终操作，并保证当前文件位置满足该操作的条件。
- 必须在同一轮回答中输出全部调用，每个文件恰好一项。同一个工具用于多个文件时，也要分别列出，不能把多个文件名拼成一个参数。
- 各文件相互独立，没有“先恢复同一个文件，再删除它”这样的前后依赖。改口是修改最终选择，不表示要把新旧操作都执行一遍。
- 调用数组的顺序不影响得分；按输入顺序列出即可。不能遗漏、增加、重复调用，也不能对同一文件输出相互冲突的操作。

输出格式

只输出一个合法 JSON 数组，允许换行和缩进。每项恰好包含 `tool` 和 `argument` 两个字符串字段，不加解释或代码块。单个 JSON 对象不代表完整的一组同步调用。

评测方式：平台先解析模型输出的 JSON，再比较数组中各个 JSON 对象的内容，不进行原始文本的逐字比较。换行、缩进、JSON 结构中的空格和对象字段顺序不同不会影响得分；字符串字段的内容（如工具名、文件名）仍须准确一致。

例如输入“删除《临时.txt》;《课程表.pdf》归档保留”，应输出：

```
[{"tool":"trash_file","argument":"临时.txt"},{"tool":"archive_file","argument":"课程表.pdf"}]
```

数组内各项顺序、对象字段顺序及 JSON 结构空格不影响得分，字符串内容必须完全一致。本题评测工具调用指令，不实际执行文件操作；同步指同一轮提交一组独立调用，不要求考生编写并发代码。

提交与评分

只提交 System Prompt 和 thinking 开关。**System Prompt 正文总共不超过 300 Token**，包括工具说明、规则和自行添加的示例。平台用固定 Qwen3-0.6B 分词器对 prompt.strip() 计数，不加特殊 Token。聊天模板、当前测试输入和模型输出不计入这 300 Token。

平台只发送你的 system Prompt 和当前 user 请求，不额外注入工具表，不使用原生 tools 参数。作答区 Prompt 初始为空，thinking 可以开启或关闭。

公开 5 点、隐藏 5 点，每点 10 分，满分 100 分。**一个测试点要求整组调用全部正确才得 10 分，不按调用条数给部分分。** 少一项、多一项、重复项、工具错误、参数错误、同文件冲突调用、额外字段、重复

JSON 字段、解释或代码块均不得分。只评分 thinking 后的最终答案，生成截断或没有最终答案不得分。

固定生成参数：temperature=0、top_p=1、max_tokens=4096、presence_penalty=0、frequency_penalty=0、stream=false。max_tokens 包含 thinking 与最终输出，与 Prompt 的 300 Token 上限分开计算。

5 个公开测试点

P01

输入：

```
删除《临时名单.xlsx》;《旧课程表.pdf》归档保留，以后还要查。
```

期望输出：

```
[{"tool":"trash_file","argument":"临时名单.xlsx"},{"tool":"archive_file","argument":"旧课程表.pdf"}]
```

P02

输入：

```
《废弃草稿.docx》永久删除，不需要恢复；从回收站找回《班会照片.zip》。
```

期望输出：

```
[{"tool":"delete_file","argument":"废弃草稿.docx"},{"tool":"restore_file","argument":"班会照片.zip"}]
```

P03

输入：

```
把《草稿_A.txt》和《草稿_B.txt》删除;《课堂笔记.md》归档留存。
```

期望输出：

```
[{"tool":"trash_file","argument":"草稿_A.txt"},{"tool":"trash_file","argument":"草稿_B.txt"},{"tool":"archive_file","argument":"课堂笔记.md"}]
```

P04

输入：

《临时表.csv》普通删除；《作废记录.txt》彻底删除，不留恢复机会；恢复回收站里的《合照.zip》；《教学计划.pdf》归档备查。

期望输出：

```
[{"tool":"trash_file","argument":"临时表.csv"},{"tool":"delete_file","argument":"作废记录.txt"},{"tool":"restore_file","argument":"合照.zip"},
{"tool":"archive_file","argument":"教学计划.pdf"}]
```

P05

输入：

先永久删除《清理说明_V2.txt》。不，取消它的永久删除，只普通删除，保留恢复机会；《活动安排.docx》归档保留。

期望输出：

```
[{"tool":"trash_file","argument":"清理说明_V2.txt"},
{"tool":"archive_file","argument":"活动安排.docx"}]
```

成人组 · T2

传感器读数整理

作答前请注意

- **可以编程生成训练数据。** 你可以在本地 IDE 编写、运行 Python 脚本，批量生成 `train.jsonl`，也可以手工准备。生成后要自行核对格式、标签与数据限制；生成脚本和提交的 `process_data` 函数用途不同。
- **珍惜训练机会：每人最多 6 次，同时只能运行 1 次。** 已启动的训练、主动取消和超时都会消耗机会；平台确认的系统故障退还机会。请先完成函数自测和数据检查，再启动训练；这两项检查不占训练机会。
- **单次训练时限 10 分钟，不设置累计训练时限。** 从训练进程启动起计，包含模型加载、训练和保存，不包含排队及训练前数据检查。每次数据限 1～60 条、256 KiB，每条完整训练消息最多 192 Token。
- **确认最终提交前选好模型。** 可选择此前任意一次成功训练的模型。确认最终提交后，本题锁定，不再训练或修改。现场只查看公开反馈，隐藏点在锁定后统一评测。

任务背景

设备每批上报 3～4 条整数读数。负数表示无效采样，同一个有效读数连续出现时只保留一次。请准备少量标注数据并训练 Qwen3-0.6B，把每批读数整理成监控页面展示的 JSON 整数数组。

你需要完成的工作

1. 在 `submission.py` 中完成 `process_data(records)`，把每条标注记录转换成训练消息。
2. 在 `train.jsonl` 中准备 1～60 条训练数据，每行一个 JSON 对象。
3. 在 `training_params.json` 中选择五项训练参数，使用平台训练功能训练模型。
4. 根据公开测试反馈修改数据或参数，最终提交一次成功训练得到的模型。

`sample_dataset.jsonl` 只展示格式，不是完整训练集。请自行检查 JSON 格式、标签正确性、规则覆盖范围和正式测试输入重合。

一、固定处理规则

输入形如 `{"readings":[2,-1,2,3]}`，数组长度为 3～4，元素为 −2～3 的整数。先删除所有负数；再对删除后的数组合并连续相同的有效读数，每段保留一个；最后保留其他读数的原顺序。`0` 是有效读数，被其他有效读数隔开的相同数值要分别保留。

例如：

- `{"readings":[2,-1,2,3]}` → `[2,3]`
- `{"readings":[1,-2,1,0]}` → `[1,0]`
- `{"readings":[0,3,1]}` → `[0,3,1]`
- 全是负数时输出 `[]`

最终只输出 JSON 整数数组，不要解释、代码块、字符串数字或布尔值。

二、完全公开的固定推理 Prompt

```
你负责整理传感器上报的读数。输入是一个 JSON 对象，readings 字段为整数数组。
按下面的固定顺序处理：
1. 负数表示无效采样，删除所有负数。
2. 对删除后的数组合并连续相同的数值，每段相同数值只保留一个。
保留其余读数的原顺序。相同数值如果被其他有效读数隔开，则分别保留。0 是有效读数。
仅输出处理后的 JSON 整数数组，不输出代码块或解释。
```

平台使用非思考模式、确定性解码和最多 64 个新 Token；训练时会自动加入同一条 system 消息。

三、训练数据与函数

`train.jsonl` 每个非空行包含字符串字段 `input`、`output`，例如：

```
{"input":"{\"readings\":[2,-1,2,3]}","output":"[2,3]"}
```

在 `submission.py` 中实现：

```
def process_data(records):
```

返回列表中每条记录只有 `messages` 字段，按原顺序放入 user、assistant 两条消息；原样保留两个字符串、空白、重复记录；忽略其他字段；不要加入 system 消息；不要读写文件或访问网络。函数可使用基本 Python，以及 `json`、`math` 标准库。代码最多 16000 个字符。

两个公开函数测试 F01、F02 在 `processing_tests.json`。

四、数据额度与重合限制

- 每条固定 system、user、assistant 和聊天模板后的完整训练消息最多 192 Token，超出直接拒绝，不截断；
- 训练输入与任一正式模型测试输入在删除全部空白后完全相同，会被拒绝；不做语义相似度筛查，也不限制训练集内部重复；

- 错误标签不会由平台修正，可能影响训练效果。

五、训练参数

`training_params.json` 只包含以下五项：

参数	默认值	可选范围
`epochs`	1	1～24 的整数
`learningRate`	0.0001	0.00001～0.0005
`loraRank`	16	4、8、16
`batchSize`	4	2、4
`gradAccum`	1	1、2、4、8、16

其余模型、Prompt、推理和优化器设置由平台固定。

六、公开测试与评分

模型公开测试输入和答案在 `public_tests.json`；隐藏测试不展示。每个模型测试点只有一条输入，答案数组内容和顺序完全正确才得分。

- F01、F02：函数点各 10 分；
- P01～P04：公开模型点各 10 分；
- H01～H04：隐藏模型点各 10 分；
- 总分 100 分，隐藏点在最终提交锁定后统一评测。

数据处理函数的公开测试通过，但训练因为函数或数据的问题没有正常完成的，只得到数据处理函数成绩分，模型部分计 0 分。可选择标为“仅函数成绩”的完成记录最终提交。

本地函数自测

在 IDE 中打开本题的 `submission.py`，填写 `process_data` 后直接运行该文件即可。

成人组 · T3

Rich 按需导入

提交与用量

这是一个 **Vibe Coding 题目**，通过对话指导 Agent 修改代码，不要求事先熟悉项目使用的编程语言。

模型：**Qwen3.5-35B-A3B**；Agent：**极简文件读写与命令执行 Harness**。

- **节约 Token，不要把整份题面一次性粘贴给 Agent。** 请认真阅读完整题面和项目材料，理解功能之间的联系，自行定位需要调整的实现，再用自己的话说明目标和关键约束。让 Agent 按需查阅相关代码，避免反复输出大段文件；输入中的对话历史、工具定义和工具返回可能在后续请求中再次发送，并再次计入 Token。
- **只能通过与 Agent 对话修改代码，不能直接编辑文件或操作终端。** 可以让 Agent 编写代码、执行命令和自检；不要求你事先熟悉 Python，也不需要你手动运行评测。
- **第一题和第二题确认提交后，才能获得本题模型 API。** 请先检查并确认前两题的最终版本。
- **留意累计用量和平台显示的剩余次数。** 一条用户消息可以触发多次模型请求与工具操作，但只计一轮用户消息；全题最多 80 轮。完成后及时提交；达到归一化用量上限 **B = 600,000** 后停止请求模型和执行工具，并自动锁定最后保存的代码版本。
- **每轮对话结束后由平台自动运行公开测试，锁定后不能再修改。** 公开 5 点现场反馈，隐藏 5 点考试结束后统一评测，每点 10 分；用量扣分在功能分基础上计算。

平台记录 Agent 的对话和工具调用。本题以**归一化输入 Token**统计用量：普通（未缓存）输入每个计 1，缓存命中输入每个计 0.1，输出每个计 8。缓存输入量按平台估算值计量：同一考生当前完整输入与此前成功请求输入的最长相同 Token 前缀按缓存计费，考试期间假设缓存不失效，不与其他考生共享。下文额度和扣分均采用此口径。

```
单次归一化用量 = (输入总 Token − 缓存输入 Token) + 0.1 × 缓存输入 Token + 8 × 输出 Token
累计用量 T = 所有请求的归一化用量之和
```

例如一次请求输入 1,000 Token，其中缓存命中 600，输出 100，则计入 400 + 60 + 800 = **1,260 个归一化输入 Token**。缓存输入已经包含在输入总量中，不重复计入普通输入；thinking如有消耗，已包含在输出中，不另外叠加。

本题用量参数：A = 300,000，B = 600,000，K = 20%。

设功能分为 `F`（满分 100），累计归一化用量为 `T`：

```
T ≤ 300,000                 ：最终分 = F
300,000 < T < 600,000   ：最终分 = F × [1 − 0.8 × (T − 300,000) / 300,000]
T ≥ 600,000                 ：停止请求与工具执行，自动锁定最后版本；最终分 = F × 20%
```

A 是不扣分额度，B 是用量硬上限，K 是衰减的最低保留比例。从 A 到 B 全程线性衰减，接近 B 时保留比例接近 20%；达到 B 后停止请求与工具执行，并自动锁定最后保存的代码版本；最终功能分保留 20%，曲线在 B 处结束。

累计归一化用量	功能分保留比例	功能分为 100 时的最终分
不超过 300,000	100%	100
375,000	80%	80
450,000	60%	60
525,000	40%	40
接近 600,000	接近 20%	接近 20
达到 600,000	20%，自动锁定最后版本	20

累计用量按上述权重计算，保留 0.1 Token 精度；最终得分四舍五入到两位小数。请求前按全部输入×1和最大输出×8预留额度，回执返回后按实际缓存和输出结算；剩余额度不足以预留下一次请求时，本次模型请求不会发出。

作答方式

你通过和代码 Agent 对话完成任务。你不需要事先熟悉这个项目使用的全部 Python 代码，也不能直接编辑工作区文件；可以让 Agent 读取、修改文件并执行命令。题目只需要命令行验证，不需要浏览器。

请认真阅读全部要求与项目说明，结合仓库目录、搜索和调用关系自行定位实现。题面不提供待修改文件清单；你可以指导 Agent 先调查依赖关系，再制定修改计划、逐项实施并检查。

背景

工作区是开源终端渲染库 Rich 的一个固定版本。部分功能只在真正使用时才需要读取较大的配置、emoji 或运行时反射模块；如果在普通导入阶段就加载这些依赖，会增加命令行程序的启动时间。

请减少下列功能在普通导入时不必要的依赖加载，同时保持现有公开 API、返回结果与异常行为兼容。类型标注专用依赖可以只在类型检查时加载；实际功能仍须正常运行。所有判断均针对一个全新的 Python 进程，不能假定其他功能已经提前加载了依赖。

功能要求

请结合项目说明与源码定位以下功能的实现及依赖关系，不限制必须修改几个文件：

1. **主题配置**：仅导入主题功能时不加载标准库 `configparser`；真正从文件读取主题时才加载，并保持主题解析和样式结果一致。

2. **表情标签替换**：仅导入替换功能时不加载完整 emoji 数据表；实际替换时按需加载。已知标签仍能替换，未知标签保持原文。
3. **表情对象**：仅导入表情对象功能时不加载完整 emoji 数据表；创建对象时按需查表，正常名称及错误名称的原有行为保持一致。
4. **对象渲染协议**：仅导入协议转换功能时不加载 `inspect`。本项只要求普通对象的一次协议转换和普通文本原样返回；调用转换功能时允许加载反射工具和控制台依赖，不要求整个渲染链都延迟加载。无需扩展转换规则或重构其他渲染功能。
5. **自动对象表示**：仅导入自动生成对象表示的功能时不加载 `inspect`；真正自动推导构造参数时才使用所需反射功能，参数、默认值和表示结果保持兼容。
6. **JSON 展示**：仅导入 JSON 展示功能时不加载标准库 `json`。从 JSON 字符串或 Python 数据创建展示对象时按需加载；保持已有公开 API、格式参数、Unicode 文本及无效输入的原有异常行为。
7. **布局空间分配**：仅导入比例分配功能时不加载标准库 `fractions`；开始调用分配函数后即可按需加载，不要求区分固定尺寸路径和分数计算路径。保持已有分配、缩减结果，以及舍入余量、固定尺寸和最小尺寸约束。

工作区与修改范围

你可以让模型逐一查看项目中的每个文件，检查这些文件的依赖关系，不仅查看这几个单独的文件就开始修改。

不能通过删除功能、伪造 `sys.modules`、识别测试输入或绕过检查来获得分数。依赖延迟加载之后，正常路径和异常路径都应保持项目原有行为。

验证

平台设置 **5 个公开计分组和 5 个隐藏计分组，每组 10 分，共 100 分**。一个计分组可能包含多个独立场景，组内全部通过才得该组分数，不按子场景给部分分。

公开反馈仅覆盖七项需求中的四项，**另外三项需求的计分测试完全隐藏，不在现场反馈中展示**。隐藏评测还检查按需加载、运行结果、异常行为及不同使用路径；公开全部通过不代表最终满分。每个评测场景都在新的 Python 进程中运行。

每轮对话结束后，平台自动运行公开测试并反馈；隐藏组及其子场景在考试结束后统一评测，现场不展示。你也可以让 Agent 运行项目测试或必要的命令行自检；不需要自己操作终端或上传补丁。

代码来源

本题基于 Rich 开源项目，遵循 MIT 许可证。初始提交为 `fc41075a3206d2a5fd846c6f41c4d2becab814fa`。以 Rich 的该提交为初始源码；考试环境另提供 `AGENTS.md`，保留 README 和 LICENSE，未挂载 `imgs/` 图像目录。