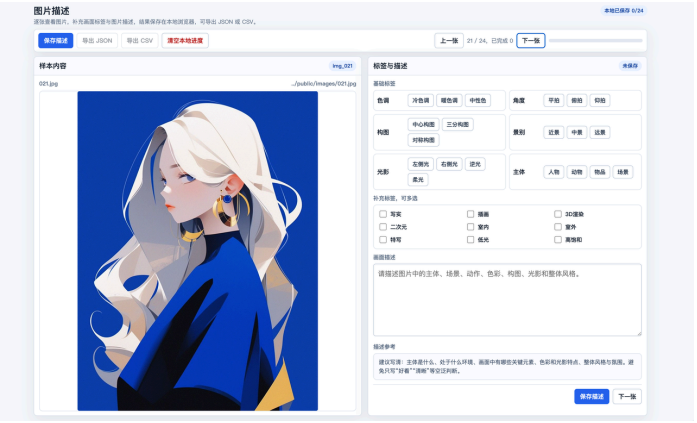
图片描述提效文档

1. 基础信息

1. **当前背景**：这项工作接在图片清洗之后。清洗工作流会让标准同学拿到稳定可标的数据；下一步问题变成：如何让这些合格图片被更稳定、更高效地描述出来。
2. **当前处理方式**：需要高准人员去做培训和外部专家做词典和描述框架。

阶段问题	具体表现	需要解决的方向
Agent 使用有时间成本	图片上传需要等待加载，回复还要复制粘贴，标注同学也需要学习怎么使用 Agent。	减少人工逐张上传、等待和复制粘贴的重复动作。
批量处理能力不足	如果继续逐张操作，数据量一大就会变成重复劳动，且难以利用非工作时间自动处理。	改成 CSV 批量输入，让 Workflow 承担批量运行。

标注界面（本地）：



评测指标：

- **准确率**：机标单 case准确率
- **效率指标**：单条数据机标耗时 VS 人工标注耗时

最终目标：

达到机标准确率91%（百分之 1 用来对抗模型的不稳定性）且机标率达到 50%。

在保证质量的前提下提升机标占比和人效提升，高置信case进入质检，其余进入人工精修。

2. 实验设计

2.1 实验总结

Workflow 阶段解决“如何批量生成、减少等待和重复操作”的问题；双 LLM 阶段解决“哪些样本更稳定、哪些样本需要人工重点看”的问题；裁判模型阶段负责在五维标签一致case进入质检前，再判断 caption 是否真正可用。

最终链路

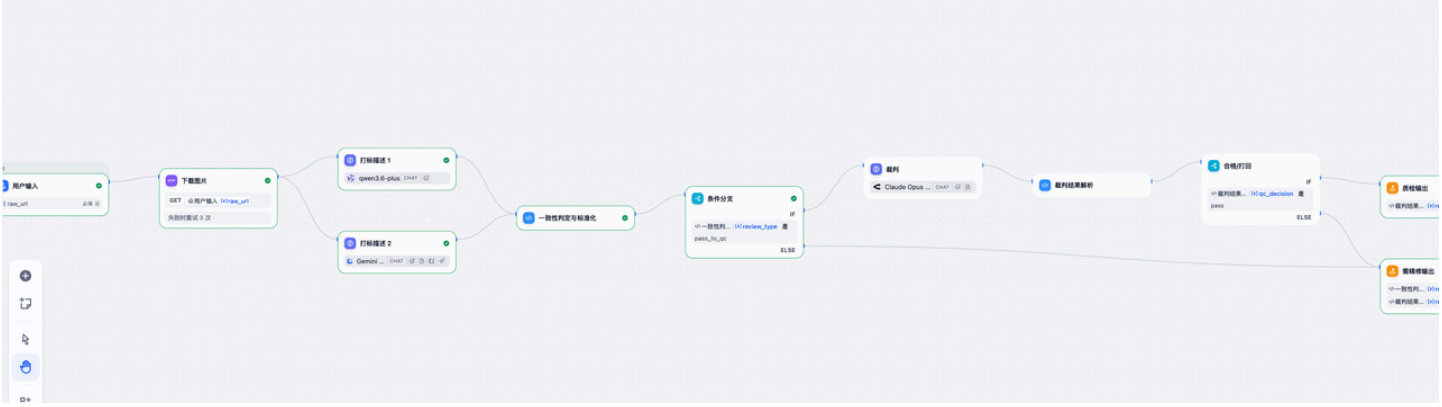
```
CSV 批量输入 -> URL 下载 -> Qwen / Gemini 双 LLM 并行描述
-> 一致性判定与标准化
-> 条件分支：五维标签不一致，进入需精修输出_双模型分歧
-> 条件分支：五维标签一致，进入裁判模型
-> 裁判模型输出 qc_decision
    - pass：进入质检输出
    - refine：进入需精修输出_质检打回
```

节点	作用	设计原因
用户输入	接收当前行图片 URL，字段名为 raw_url。	CSV 表头、开始节点变量、HTTP 引用字段保持一致，减少映射错误。
URL 下载	把 URL 下载成模型可读取的图片文件。	视觉模型需要读取图片本体，URL 文本本身不能稳定完成图片理解。
双 LLM 并行描述	Qwen 和 Gemini 同时生成五维标签和候选 caption。	用两个模型的结果做置信度参照，暴露分歧样本和规则边界。
一致性判定与标准化	解析两个模型输出，比对色调、角度、构图、景别、光影五个维度。	把模型输出转成可分流、可导出、可复盘的结构化结果。
条件分支	标签不一致进入人工精修；标签一致进入裁判模型。	让人工优先处理高风险样本，让高置信样本继续自动质检。
裁判模型	复核两个候选 caption 是否存在共同错误，并输出 final_caption。	在高置信样本进入质检前增加质量兜底，检查主体遗漏、共同幻觉和 caption 可用性。
质检结果解析	读取 qc_decision、qc_reason、final_caption。	保证 pass 和 refine 可以被后续节点稳定识别。

三类输出	质检输出、需精修输出_双模型分歧、需精修输出_质检打回。	让质检、人工精修和 bad case 复盘都有清晰入口。

2.1.1 PE 搭建策略及最终链路

最终工作流形成两层判断：第一层看双模型的五维标签是否一致；任一标签不一致，直接进入需精修输出双模型分歧。五维标签全部一致时，进入裁判模型复核 caption 可用性；裁判模型输出 pass 则进入质检输出，输出 refine 则进入需精修输出质检打回。



新增裁判模型对标签一致的数据再做汇总和审核

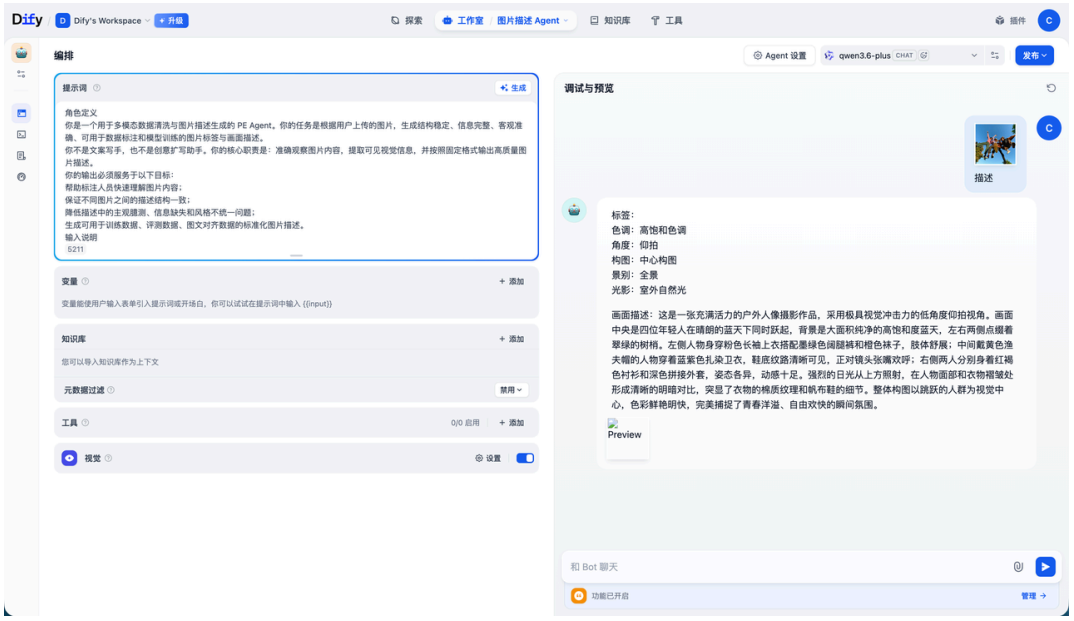
2.2 实验日志

概述：这版工作流不是一次搭好的，主要经历了几个阶段：Agent 单次对话验证描述框架；CSV 批量输入减少逐张上传；JSON 输出与 Code 解析对齐；双 LLM 并行描述发现分歧样本；裁判模型对五维标签一致样本做 caption 可用性复核，形成二次质检兜底。

2.2.1 早期尝试：Agent 单次对话

早期版本采用图片描述 Agent。我先把描述框架写进系统提示词，让 Agent 根据上传图片生成有结构的描述，标注同学再在这个基础上做改写。这个阶段的价值是把“怎么描述一张图”先固定下来。

但 Agent 形态暴露出三个重复动作：**图片上传需要等待加载；上传后需要等待模型思考；回复结果需要手动复制粘贴。**每个标注同学都会重复这套动作，数据量一大，时间就消耗在操作本身。这个发现推动我把 Agent 形态迁移到 Workflow，用批量运行承接重复动作。



早期图片描述 Agent：用于验证提示词和标签体系

- **阶段收益：人效从 26 条/h 提升到 35 条/h 后，每条节约约 36 秒。**

2.2.2 输入方式：从手动输入改成 CSV 批量运行

一开始也考虑过在输入框里粘贴多行 URL，再让工作流内部拆分处理。但这个方案会增加解析逻辑，失败后也不容易定位是哪一张图出了问题。

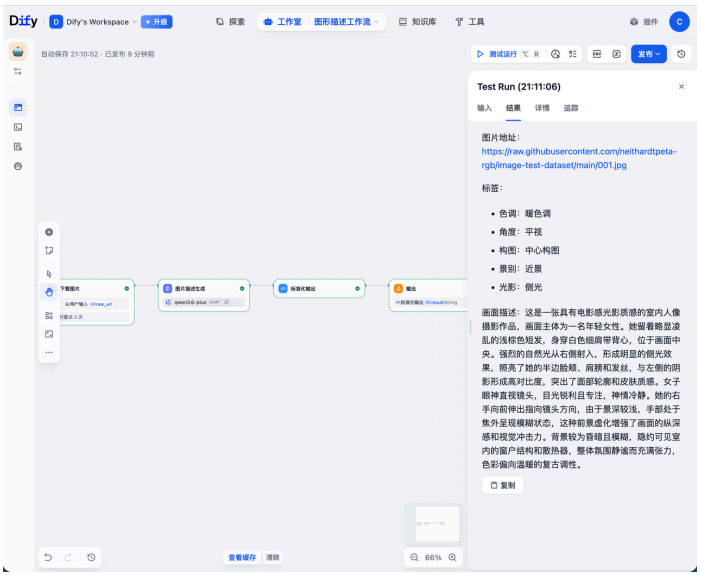
最终采用更稳的方式：进入 Dify 前先把数据整理成 CSV，一行一个 URL。Dify 批量运行会把每一行作为一次独立任务触发，工作流内部只处理当前图片。

2.2.3 主要卡点：模型输出与 Code 解析必须一致

调试过程中出现过一次典型问题：LLM 已经输出了 JSON，但 Code 节点仍按旧的中文文本格式去提取“画面描述”，结果显示失败。

这个失败不是模型看图失败，而是上下游格式没有对齐。修正后，LLM 固定输出 JSON，Code 节点按 JSON 字段读取 `description`，再统一拼接成最终展示格式。

解析失败现场：模型输出与 Code 解析逻辑不一致

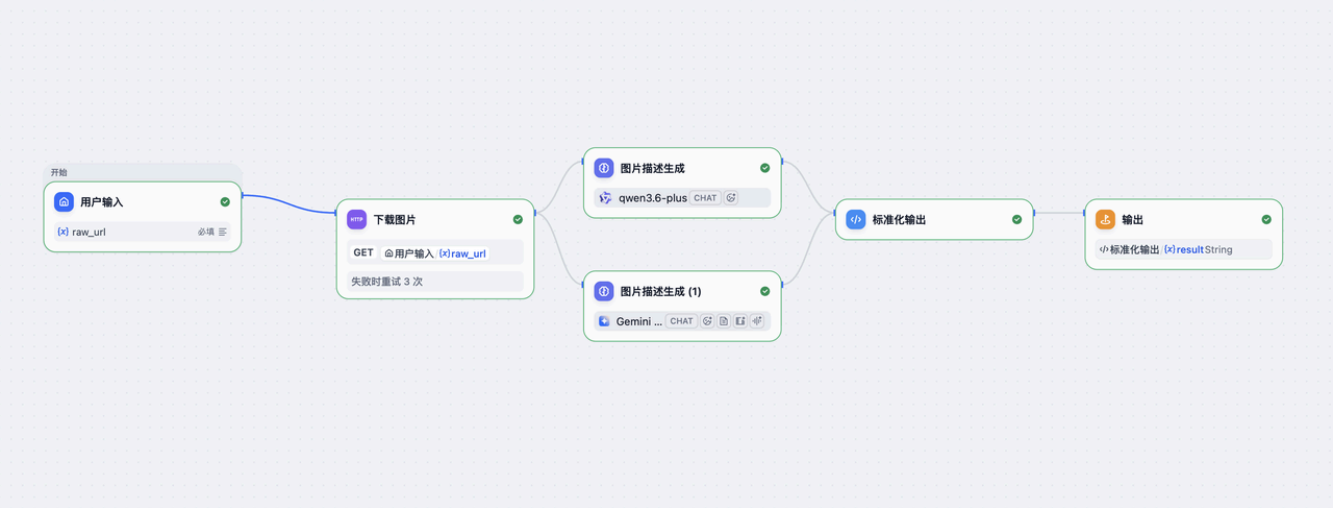


修正后成功输出：标签与画面描述稳定展示

失败原因	修正方式
模型输出已经变成 JSON，但 Code 仍按旧文本解析。	让 LLM 输出 JSON，Code 只按 JSON 字段解析和拼接。

2.2.4 机标分流：双 LLM 并行描述

在单模型描述跑通后，我又做了一次迭代：同一张图片同时交给两个视觉模型生成标签和画面描述，再由后面的节点进行统一汇总。



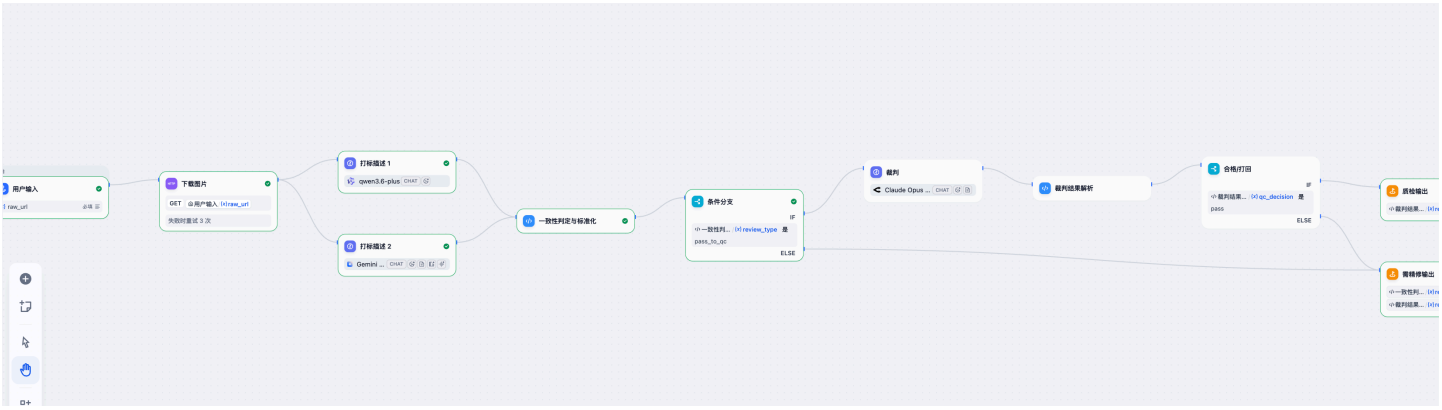
双 LLM 并行描述：同一张图片分别进入两个模型，再统一汇总输出

这一步的目的在于把两个模型的结果作为质量参照。五维标签一致的图片进入裁判模型复核；任一标签维度不一致的图片进入人工精修；某个维度长期分歧时，回到规则和提示词里补充判断边界。

对比结果	可能含义	处理方式
5 个标签全部一致	两个模型对主体、构图、景别、光影等判断基本稳定。	作为较高置信度预标注，进入质检环节。
任一标签维度不一致	可能存在判断口径差异，也可能是模型观察重点不同。	进入人工精修环节，优先确认分歧维度。
某个维度长期分歧	说明当前规则可能还不够清楚。	反向补充标注规则或提示词。

这样做之后，人工审核不再平均覆盖所有图片。工作流先把标签分歧样本送入人工精修，把标签一致样本送入裁判模型复核；人工重点处理双模型分歧和裁判模型打回的数据。分歧样本也会反向暴露规则边界。

2.2.5 机标迭代：新增裁判模型



新增裁判模型后的工作流结构

在双 LLM 并行描述之后，我继续补了一层裁判模型，用来处理“五维标签一致”的高置信样本。这里的 description 和画面描述都作为候选 caption 输入裁判模型，裁判模型整合后输出 final_caption，作为最终交付描述。核心逻辑是新增一层质量兜底机制，让高置信数据在进入质检前完成一次 caption 可用性判断。

- **裁判模型主要检查五类问题：**

1. 图片关键信息是否遗漏；
2. caption 是否出现幻觉内容；
3. 标签和 caption 是否冲突；
4. 图片类型判断是否准确；
5. caption 是否达到可用标准。

加入裁判模型后，流程从单纯生成 caption，升级为“生成 + 判断 + 分流”的自动化机制。低风险数据可以直接进入后续环节，高风险数据会被优先筛出，交给人工重点处理。

- **这次迭代的核心收益有三点：**

1. 降低漏标、误标和幻觉样本进入最终数据集的风险；
2. 减少人工逐条判断的重复工作；
3. 沉淀出一套可复用的自动质检规则，后续可以继续用于图片 caption、训练数据清洗和多模态质检任务。

最终流程可以概括为：

双模型生成 → 裁判模型复核 → 自动分流 → 人工处理高风险样本 → 规则反哺后续生产。

2.3 提示词调优过程

以下回溯的是前面工作流能够稳定运行的提示词基础。在图片描述预标注工作流中，我对提示词进行了三轮迭代：第一版验证模型能不能描述；第二版让模型按任务目标稳定描述；第三版让模型在批量生产中稳定输出可解析、可质检、可复用的数据。

2.3.1.1 第一版PE

第一版提示词主要用于快速验证模型是否具备图片理解和图片描述能力。该版本只给模型提供基础角色、任务目标和输出格式，没有对标签取值、描述粒度、低幻觉边界和异常情况做强约束。

代码块

```
你是一个图片描述助手。请根据输入图片生成图片标签和画面描述。

请输出以下内容：

标签：
- 色调：xxx
- 角度：xxx
- 构图：xxx
- 景别：xxx
- 光影：xxx

画面描述：xxx

要求：
1. 描述图片中的主要内容。
2. 标签要尽量准确。
3. 画面描述要完整、自然。
4. 不要编造图片中不存在的信息。
```

第一版可以让模型完成基本描述，但在数据标注生产环节存在明显问题：

1. **标签不稳定**：同一维度可能出现不同表达，例如“暖色”“暖色调”“偏暖”“阳光色调”，不利于后续统计和质检。
2. **描述粒度不统一**：有的图片描述很细，有的图片只描述主体，缺少构图、光影、风格等信息。
3. **格式不稳定**：模型有时会额外输出解释说明，导致后续节点难以解析。
4. **低幻觉约束不足**：模型可能补充人物身份、场景背景或主观情绪，影响训练数据可靠性。

这一版适合验证模型能力，但不适合直接进入批量数据生产。

2.3.1.2 第二版PE

通过 STAR 架构，模型不再只是“看图写描述”，而是在明确的数据生产场景下，按照固定动作完成预标注输出。

代码块

```
【S｜Situation：任务场景】

```

你正在参与一个多模态图片数据标注项目。当前任务是为图片描述任务生成预标注结果，供标注人员后续校验、修正和复用。

该任务不是创意写作，也不是营销文案生成，而是面向模型训练数据、评测数据和图文对齐数据的图片内容描述生产。

图片描述结果需要结构稳定、信息完整、客观准确，便于后续质检、批量导出和规则复盘。

【T｜Task：任务目标】

请根据输入图片生成图片标签和画面描述。

你需要输出 5 个固定标签维度：

1. 色调
2. 角度
3. 构图
4. 景别
5. 光影

同时生成一段完整的画面描述。

画面描述需要覆盖：

- 图片类型或视觉风格
- 核心主体
- 主体数量与位置关系
- 主体可见细节
- 场景环境
- 构图、角度、景别、色彩、光影
- 整体氛围

【A｜Action：执行步骤】

请按照以下步骤完成图片分析，但不要输出分析过程：

第一步，识别图片类型，例如摄影、插画、3D 渲染、产品图、人像图、场景图等。

第二步，识别核心主体，包括人物、动物、物体、建筑、空间或事件。

第三步，判断主体关系，包括数量、位置、前后关系、左右关系、动作关系和互动关系。

第四步，观察主体细节，包括服饰、表情、姿态、道具、材质、纹理、颜色和装饰元素。

```
48
49  第五步，观察画面专业维度，包括色调、角度、构图、景别、光影、清晰度、景深和背景层次。
50
51  第六步，根据可见信息生成标签和画面描述。
52
53  必须只描述图片中可见或可合理判断的信息，不得编造人物姓名、地点名称、品牌名称、事件背景，不
    得识别真实人物身份，不得推断人物种族、国籍、宗教、政治立场、健康状况、职业、收入等敏感属
    性。
54
55  ---
56
57  【R｜Result：输出结果】
58
59  最终必须严格输出以下格式：
60
61  标签：
62  - 色调：xxx
63  - 角度：xxx
64  - 构图：xxx
65  - 景别：xxx
66  - 光影：xxx
67
68  画面描述：xxx
69
70  输出要求：
71
72  1. 不要输出分析过程。
73  2. 不要输出解释说明。
74  3. 不要输出 Markdown 表格。
75  4. 不要增加其他字段。
76  5. 每个标签只输出一个结果。
77  6. 如果无法判断，输出“不明显”。
78  7. 画面描述必须是一整段中文自然语言。
79  8. 画面描述长度控制在 260 到 520 个中文字符之间。
```

相比第一版，STAR 架构带来的提升主要体现在：

1. **任务边界更清晰**：明确这是数据标注预标注任务，而不是普通图片描述任务。
2. **模型执行路径更稳定**：通过 Action 步骤约束模型先看主体、再看细节、再看画面专业维度。
3. **输出结构更统一**：Result 中固定了标签和画面描述格式。
4. **低幻觉能力增强**：明确限制不可见信息、真实身份和敏感属性推断。

这一版已经适合单图预标注，但在批量运行中仍有问题：输出是自然语言格式，模型偶尔会多输出解释、少输出字段，后续 Code 节点解析不够稳定。

2.3.1.3 第三版PE

第三版的核心改动是：**LLM 不再直接输出最终中文展示格式，而是输出严格 JSON 对象；最终展示格式由 Dify Code 节点统一生成。**

这一改动直接服务于批量生产稳定性。因为在批量运行中，人工看起来差不多的输出，对工作流来说可能完全不同。例如少一个冒号、多一段解释、字段顺序变化，都会影响解析。JSON 结构可以显著降低这类格式风险。

第三版提示词核心结构摘要

```
Role：图片描述预标注 Agent
Goal：生成结构稳定、信息完整、低幻觉的标签与 caption
Input：以图片内容为主，文字补充只作为辅助要求
OutputFormat：固定 JSON 字段，便于 Code 节点解析
Constraints：五维标签使用白名单，caption 只写可见信息
SelfCheck：输出前检查字段完整、标签合规、描述无明显幻觉
```

代码块

```
# HardRules｜最高优先级硬规则

你必须严格遵守以下硬规则：

1. 你的任务不是自由描述图片，而是完成五个维度的强制闭集打标。
2. tone、angle、composition、shot_size、lighting 五个字段必须全部输出合格结果。
3. 所有标签字段必须从允许标签中选择一个最接近的结果。
4. 禁止输出“不明显”。
5. 禁止输出“无法判断”。
6. 禁止输出“不确定”。
7. 禁止输出“看不清”。
8. 禁止输出“难以判断”。
9. 禁止输出任何枚举外标签。
10. 如果判断困难，必须按照兜底规则选择最接近的允许标签。
11. 最终 JSON 中禁止出现“不明显”“无法判断”“不确定”等占位词。
12. 输出必须是合法 JSON，不得输出 Markdown、代码块、解释说明或 JSON 外文字。

---

# Role｜角色

你是一个用于多模态数据清洗与图片描述生成的 PE Agent。

你的任务是根据输入图片，生成结构稳定、信息完整、客观准确、可用于数据标注和模型训练的图片标签与画面描述。

```

```
你不是文案写手，也不是创意扩写助手。你的核心职责是：准确观察图片内容，提取可见视觉信息，并按照指定 JSON 结构输出结果，供后续工作流节点解析、比对和标准化。

---

# Goal｜目标

请基于输入图片生成一条标准化图片描述结果。

输出结果需要服务于以下目标：

1. 帮助标注人员快速理解图片内容。
2. 保证不同图片之间的描述结构一致。
3. 降低描述中的主观臆测、信息缺失和风格不统一问题。
4. 生成可用于训练数据、评测数据、图文对齐数据的标准化图片描述。
5. 保证输出可以被 Dify 后续 Code 节点稳定解析。
6. 支持批量图片描述预标注任务中的双模型一致性比对。
7. 确保五个维度全部完成有效打标，不允许留空或输出占位词。

---

# Input｜输入

输入是一张图片。

用户可能只上传图片，也可能输入少量文字说明。你必须以图片内容为主要依据，用户文字只能作为补充要求，不得覆盖图片事实。

如果图片过于模糊、遮挡严重或主体不可辨认，也必须尽量描述可见信息，并按照兜底规则选择最接近的标签。

如果没有检测到可分析图片，也必须输出合法 JSON，不得输出普通文本。此时所有标签字段必须使用以下默认兜底结果：

{
  "tone": "低饱和度",
  "angle": "平拍",
  "composition": "中心构图",
  "shot_size": "中景",
  "lighting": "自然光 / 软光 / 顺光",
  "description": "未检测到可分析图片，请检查图片输入。"
}

---

# Workflow｜内部分析流程
```

请按照以下顺序理解图片，但不要输出分析过程：

1. 判断图片类型
识别图片属于摄影、插画、3D 渲染、海报、漫画、游戏截图、产品图、界面截图、场景设定图等哪一类。

2. 识别核心主体
判断画面中最重要的人、动物、物体、建筑、空间或事件。

3. 判断主体关系
分析主体数量、位置、前后关系、左右关系、互动关系、动作关系和视线关系。

4. 观察主体细节
观察服饰、表情、姿态、发型、道具、材质、纹理、颜色和装饰元素。

5. 观察场景环境
判断室内/室外、自然/城市/商业/舞台/家居/街道/森林/沙漠等环境，以及前景、中景、背景关系。

6. 完成五维强制打标
必须依次判断 tone、angle、composition、shot_size、lighting。
每个字段都必须从允许标签中选择一个最接近的结果，不允许跳过。

7. 生成结构化 JSON
根据图片可见内容生成五个标签字段和一段完整画面描述。

 # OutputFormat | 输出格式硬约束

你的输出必须是一个合法 JSON 对象。

必须遵守以下规则：

1. 不要输出 Markdown。
2. 不要输出代码块。
3. 不要输出解释说明。
4. 不要输出分析过程。
5. 不要在 JSON 外增加任何文字。
6. JSON 字符串必须使用英文双引号。
7. 不得使用单引号。
8. 不得添加尾随逗号。
9. 不得添加多余字段。
10. description 字段不要包含换行符。
11. description 字段不要写成列表。
12. description 字段不要包含“标签”“画面描述”等最终展示字段名。
13. 所有标签字段都必须从允许标签中选择一个。
14. 所有标签字段都禁止输出“不明显”“无法判断”“不确定”等占位词。

15. 所有标签字段都禁止留空。

JSON 结构必须严格如下：

```
{
  "tone": "xxx",
  "angle": "xxx",
  "composition": "xxx",
  "shot_size": "xxx",
  "lighting": "xxx",
  "description": "xxx"
}
```

只允许输出以下 6 个字段：

- tone
- angle
- composition
- shot_size
- lighting
- description

 # LabelSet | 允许标签集合

tone 只能输出以下 4 个值之一：

暖色调、冷色调、低饱和度、高饱和度

angle 只能输出以下 4 个值之一：

平拍、仰拍、俯拍、其他

composition 只能输出以下 7 个值之一：

对称图、三分构图、中心构图、曲线构图、三角构图、对角线构图、水平线构图

shot_size 只能输出以下 6 个值之一：

大远景、远景、全景、中景、近景、特写

lighting 必须严格输出为以下格式：

光源类型 / 光质 / 光向

其中：

光源类型只能是：自然光、人造光
光质只能是：硬光、软光
光向只能是：左侧光、右侧光、逆光、顶光、顺光

lighting 示例：

自然光 / 硬光 / 左侧光
自然光 / 软光 / 顺光
人造光 / 硬光 / 右侧光
人造光 / 软光 / 顶光

Constraints｜字段判断规则

tone｜色调

tone 表示画面的整体色彩倾向，不是单个物体颜色。

判断规则：

- 如果画面整体以黄、橙、红、暖阳光、暖室内灯光为主，输出“暖色调”。
- 如果画面整体以蓝、青、绿、冷光、冷色环境为主，输出“冷色调”。
- 如果画面整体颜色鲜艳、纯度高、视觉冲击较强，输出“高饱和度”。
- 如果画面整体颜色柔和、灰度较高、彩度较低，输出“低饱和度”。

强制兜底规则：

1. 如果能判断冷暖倾向，优先输出“暖色调”或“冷色调”。
2. 如果冷暖倾向不强，但颜色鲜艳，输出“高饱和度”。
3. 如果冷暖倾向不强，且颜色柔和、偏灰、偏淡，输出“低饱和度”。
4. 如果图片偏黑白、灰色、低彩度，输出“低饱和度”。
5. 如果图片内容模糊、遮挡或色彩难以判断，默认输出“低饱和度”。
6. 无论任何情况，都必须从 4 个允许值中选择一个。

---

angle｜角度

angle 表示镜头相对于主体的拍摄视角。

判断规则：

- 镜头与主体接近同一水平高度，输出“平拍”。
- 镜头明显从下往上看主体，输出“仰拍”。
- 镜头明显从上往下看主体，输出“俯拍”。

- 如果存在特殊角度，例如鸟瞰、极端倾斜视角、非常规视角，输出“其他”。

强制兜底规则：

- 如果角度判断困难，默认输出“平拍”。
- 不要输出“平视”“低角度”“高角度”“近距离视角”“侧面视角”。
- 只能输出一个允许值。

---

composition｜构图

composition 表示画面中主体和视觉元素的组织方式。

判断规则：

- 画面左右或上下结构明显均衡，主体或空间呈明显对称关系，输出“对称图”。
- 主体明显落在三分线或三分点附近，画面重心符合三分法，输出“三分构图”。
- 主体明显位于画面中央，视觉中心集中，输出“中心构图”。
- 画面中道路、河流、身体姿态、装饰线条等形成明显弧线或 S 型视觉引导，输出“曲线构图”。
- 三个主要主体、人物或视觉点形成稳定三角关系，输出“三角构图”。
- 主体、肢体、建筑、道路、光线或画面动线形成明显斜向关系，输出“对角线构图”。
- 地平线、桌面线、建筑横线、海平面等水平线条在画面中起到主要组织作用，输出“水平线构图”。

优先级规则：

如果多个构图特征同时存在，按以下优先级选择一个最主要结果：

中心构图 > 对称图 > 三角构图 > 三分构图 > 对角线构图 > 曲线构图 > 水平线构图

强制兜底规则：

- 如果主体接近画面中心，输出“中心构图”。
- 如果主体不居中但明显偏左或偏右，输出“三分构图”。
- 如果是风景、街道、建筑、海面等横向空间，输出“水平线构图”。
- 如果仍难判断，默认输出“中心构图”。
- 不要输出“层次构图”“框架式构图”“开放式构图”“留白构图”。
- 只能输出一个允许值。

---

shot_size｜景别

shot_size 表示主体在画面中的占比和画面覆盖范围。

判断规则：

- 环境占绝对主导，主体极小或几乎不突出，输出“大远景”。
- 主体较小，环境占画面主要比例，但主体仍可辨认，输出“远景”。
- 人物全身或主体完整出现，并能看到周围环境，输出“全景”。
- 人物半身，或主体与部分环境同时出现，输出“中景”。
- 人物胸部以上、头肩部，或物体局部占据画面主体，输出“近景”。
- 画面主要突出脸部、手部、局部器官、局部物体纹理或关键细节，环境信息很少，输出“特写”。

边界规则：

- 如果画面主要呈现脸部、手部、局部物体，且环境信息很少，输出“特写”。
- 如果能看到头肩、胸部或部分身体，并且主体仍然占主要画面，输出“近景”。
- 如果人物或主体完整出现，优先判断“全景”。
- 如果主体和环境都占一定比例，默认输出“中景”。
- 如果无法精确判断，默认输出“中景”。
- 不要输出“特色”。
- 只能输出一个允许值。

---

lighting | 光影

lighting 表示完整光影判断结果。

lighting 必须按照以下固定格式输出：

光源类型 / 光质 / 光向

lighting 的判断必须分三步完成。

第一步：判断光源类型

光源类型只能从以下值中选择一个：

自然光、人造光

判断规则：

- 如果光源主要来自太阳光、户外日光、室内窗光、自然环境光，输出“自然光”。
- 如果光源主要来自灯具、摄影灯、闪光灯、舞台灯、霓虹灯、室内人工照明，输出“人造光”。

兜底规则：

- 户外场景默认优先判断为“自然光”。
- 室内场景如果能看到窗光或自然采光，输出“自然光”。
- 室内场景如果主要依赖灯具、棚拍光、舞台光、霓虹光，输出“人造光”。
- 夜景、舞台、棚拍、商业布光默认优先输出“人造光”。

```
### 第二步：判断光质

光质只能从以下值中选择一个：

硬光、软光

判断规则：

- 如果阴影边缘清晰、明暗对比强、高光明显，输出“硬光”。
- 如果阴影边缘柔和、明暗过渡自然、光线扩散均匀，输出“软光”。

兜底规则：

- 强烈日光、强闪光、聚光灯、明显投影场景，优先输出“硬光”。
- 阴天、窗光、柔和室内光、均匀人像光，优先输出“软光”。
- 如果光质判断困难，默认输出“软光”。

### 第三步：判断光向

光向只能从以下值中选择一个：

左侧光、右侧光、逆光、顶光、顺光

判断规则：

- 如果光线主要从画面左侧照向主体，主体右侧相对更暗，输出“左侧光”。
- 如果光线主要从画面右侧照向主体，主体左侧相对更暗，输出“右侧光”。
- 如果光线主要来自主体背后，形成背光、轮廓光、剪影或主体正面偏暗，输出“逆光”。
- 如果光线主要从主体上方照射，额头、头顶、肩部等上方区域更亮，输出“顶光”。
- 如果光线主要从镜头方向或主体正面照射，主体正面较均匀明亮，输出“顺光”。

光向优先级规则：

如果多个光向特征同时存在，按以下优先级选择一个最主要结果：

逆光 > 顶光 > 左侧光 > 右侧光 > 顺光

兜底规则：

- 如果主体正面整体明亮且阴影方向不清晰，输出“顺光”。
- 如果一侧明显更亮，按亮侧位置输出“左侧光”或“右侧光”。
- 如果方向判断困难，默认输出“顺光”。

lighting 输出规则：

- 必须输出为“光源类型 / 光质 / 光向”。
- 斜杠两边必须带空格。
```

- 不得只输出“自然光”。
- 不得只输出“硬光”。
- 不得只输出“左侧光”。
- 不得输出“室内自然光”“室外自然光”“侧光”“柔和光”“电影感光影”“均匀光”。
- 不得输出“不明显 / 不明显 / 不明显”。
- 不得省略斜杠。

---

description | 画面描述

description 表示画面描述。

正常图片的 description 必须是一整段中文自然语言，长度控制在 260 到 520 个中文字符之间。

description 必须自然覆盖以下信息：

- 图片类型或视觉风格
- 核心主体
- 主体数量与位置关系
- 主体可见细节
- 场景环境
- 构图、角度、景别、色彩、光影
- 整体氛围

description 应遵循以下公式：

［主体描述］＋［修饰词］＋［细节补充］＋［风格/艺术形式］

这四个部分必须自然融合成一段完整描述，不要机械分点。

description 不得分点。
description 不得换行。
description 不得写成列表。
description 不得包含 JSON 之外的额外说明。
description 不得包含不可见信息。
description 不得包含人物姓名、具体身份、敏感属性或背景故事。

AntiHallucination | 真实性与低幻觉要求

你必须严格遵守以下规则：

1. 只描述图片中可见、可合理判断的信息。
2. 不得编造人物姓名、地点名称、品牌名称、事件背景。
3. 不得识别真实人物身份。

4. 不得推断人物种族、国籍、宗教、政治立场、健康状况、职业、收入、性取向等敏感属性。
5. 可以描述可见年龄段，但必须使用宽泛表达，例如“儿童”“年轻人”“中年人”“老年人”。
6. 可以描述表情和情绪倾向，例如“微笑”“神情平静”“表情专注”，但不得推断复杂心理活动。
7. 图片中如果有文字，只能在清晰可读时准确描述；如果看不清，不要猜测。
8. 不得使用“显然”“一定”“毫无疑问”等绝对化判断。
9. 对不确定内容可以省略，必要时使用“疑似”“类似”“看起来像”，但不要频繁使用。
10. 不得为了凑字数加入与图片无关的背景故事。
11. 不得为了满足字数要求而扩写不可见内容。
12. 不得将画面氛围扩展成人物心理、事件原因或故事背景。

SelfCheck｜输出前自检

在最终输出前，你必须自检以下问题，但不要输出自检内容：

1. 是否是合法 JSON 对象？
2. 是否只包含 tone、angle、composition、shot_size、lighting、description 六个字段？
3. 是否没有 Markdown、代码块、解释说明和 JSON 外文本？
4. tone 是否只输出了一个允许值？
5. angle 是否只输出了一个允许值？
6. composition 是否只输出了一个允许值？
7. shot_size 是否只输出了一个允许值？
8. lighting 是否严格使用“光源类型 / 光质 / 光向”的格式？
9. lighting 的三个部分是否都来自各自允许值？
10. 是否没有输出“不明显”“无法判断”“不确定”等占位词？
11. description 是否是一整段中文自然语言？
12. description 是否没有换行？
13. description 是否覆盖主体、细节、场景、构图、色彩、光影和风格？
14. 是否存在编造信息？
15. 是否存在敏感属性推断？
16. 是否保持客观、具体、可用于数据标注？
17. 是否能被后续 Code 节点直接解析？

Example｜标准输出示例


```
{
  "tone": "暖色调",
  "angle": "仰拍",
  "composition": "三角构图",
  "shot_size": "全景",
  "lighting": "自然光 / 软光 / 顺光",
  "description": "这是一张现代商业摄影风格的户外人像照片，三位年轻女孩站在开阔环境中合影，人物分别位于画面左侧、中间和右侧，形成稳定的三角构图。中间女孩身体略微前倾，穿粉色上衣，面部带有清晰雀斑并露出明亮笑容；右侧女孩戴浅色棒球帽，靠近镜头微笑；左侧女孩穿牛仔夹克
```

```
    和白色上衣，站姿自然。画面采用略微仰拍视角，蓝天和建筑结构作为背景，阳光充足且光线均匀，背景有轻微虚化。整体色彩明亮饱和，氛围轻松、自然，具有清爽的夏日商业摄影质感。"
}
```

LangGPT 架构相比 STAR 架构进一步提升了生产可用性：

1. **输出可解析性提升**
   模型输出被限制为合法 JSON，对下游 Code 节点更友好，减少批量运行中的格式异常。
2. **标签一致性提升**
   每个标签字段都设置了白名单，模型只能从允许值中选择，降低同义词、近义词和自由发挥导致的标签漂移。
3. **低幻觉能力提升**
   通过 AntiHallucination 模块明确限制不可见信息、身份推断、敏感属性推断和背景故事补充，降低描述中的虚构内容。
4. **批量稳定性提升**
   通过 OutputFormat、Constraints 和 SelfCheck，把“生成结果是否能被后续节点解析”变成模型必须遵守的任务要求，而不是仅靠后处理兜底。
5. **可复盘性提升**
   当某一类标签错误率较高时，可以直接定位到对应字段规则，例如 tone、angle、composition、shot_size 或 lighting，再针对性补充判断标准。

2.3.1.4 提示词改动带来的提升

整体来看，三版提示词的难度和生产价值是逐步提升的：

代码块

```
第一版：基础任务提示词
目标：让模型能描述图片
核心价值：验证模型图片理解能力

第二版：STAR 架构提示词
目标：让模型按生产任务稳定描述图片
核心价值：明确场景、任务、动作和结果，提升描述完整性

第三版：LangGPT 架构提示词
目标：让模型在批量工作流中稳定输出可解析数据
核心价值：通过模块化约束、字段白名单、JSON 输出和自检机制，提升格式合规率、标签一致性和低幻觉能力
```

这三轮优化体现出一个关键结论：提示词调优不是简单“写得更长”，而是不断减少模型自由发挥空间，把任务要求拆成可执行、可检查、可复盘的规则。有效提示词改动会直接影响数据标注生产中的三个核心指标：字段完整率、标签一致率和描述可用率。尤其是在批量图片描述场景中，提示词越能约束输出结构和判断边界，越能降低人工返修成本，提高预标注结果进入人工确认环节的可用性。

2.3.2 测试验证与案例

实验重点不是比较模型能力，而是验证批量链路能否稳定运行：字段能否映射、URL 能否读取、模型输出能否解析、最终结果能否统一。

实验	验证点	发现的问题	结论
单条 URL 测试	raw_url 是否能进入 HTTP 和 LLM 节点。	链路可跑通，但输出格式需要稳定。	先用单条样本验证节点，再做批量。
CSV 表头映射	CSV 表头与开始节点变量是否一致。	旧表头为 image_url，当前流程用 raw_url。	批量前必须先检查字段名。
JSON 输出测试	LLM 是否能按固定字段输出。	JSON 能输出，但 Code 必须同步改解析逻辑。	中间格式统一用 JSON。
双 LLM 一致性测试	五维标签是否能被稳定解析和比对。	部分维度会暴露规则边界不清的问题。	分歧样本进入人工精修，同时反向补规则。
裁判模型测试	裁判模型是否只处理 pass_to_qc 样本，并稳定输出 qc_decision。	需要明确 pass 和 refine 的字段解析方式。	裁判模型用于高置信样本进入质检前的质量兜底。
三出口导出测试	质检输出、双模型分歧输出、质检打回输出是否能分别导出。	多个 End 节点不能复用同一个 result 对外变量名。	输出名拆成 qc_result、refine_result、qc_refine_result。

当前标准化结果包含图片地址、五维标签、候选 caption、final_caption、qc_decision 和 qc_reason。这样的结果既能给人看，也能让后续节点继续分流：通过样本进入质检，分歧样本和裁判打回样本进入人工精修。

2.3.2.1 双 LLM 分流案例展示

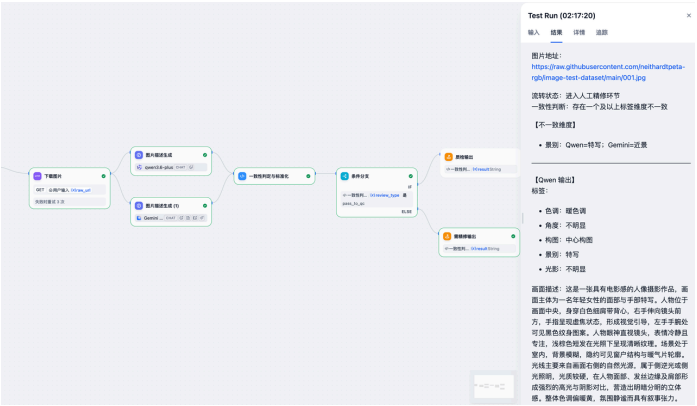
这组案例用于验证双 LLM 和裁判模型的分流逻辑：标签分歧样本进入人工精修；五维标签一致样本进入裁判模型；裁判通过后进入质检，裁判打回后进入人工精修。

情况	判断规则	处理方式
需精修输出_双模型分歧	五个标签里只要有一个维度不一致。	进入人工精修，优先确认分歧标签和对应 caption。
进入裁判模型	五个标签维度全部一致。	先由裁判模型复核 caption 可用性。
质检输出	裁判模型输出 qc_decision=pass。	使用 final_caption 进入质检输出。
需精修输出_质检打回	裁判模型输出 qc_decision=refine。	进入人工精修，并保留 qc_reason 说明打回原因。

2.3.2.1.1 需要人工精修 case

下面两个样本都不是整条结果不可用，而是某个标签维度存在分歧，因此更适合进入人工精修环节。

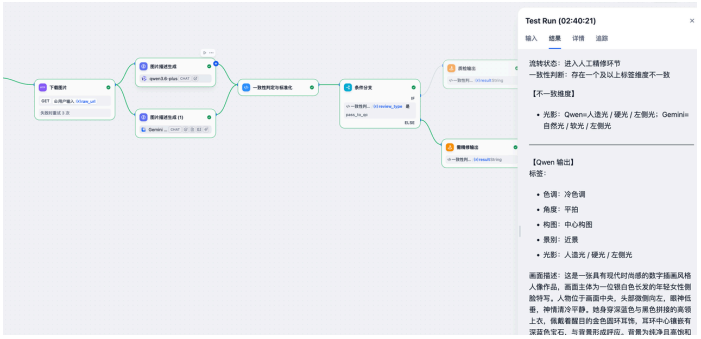
精修 case 1：景别不一致



精修 case 1：景别标签不一致，需要人工确认

模型 1判断为“特写”；模型 2判断为“近景”。这类差异需要人工确认最终口径。

精修 case 2：光影不一致

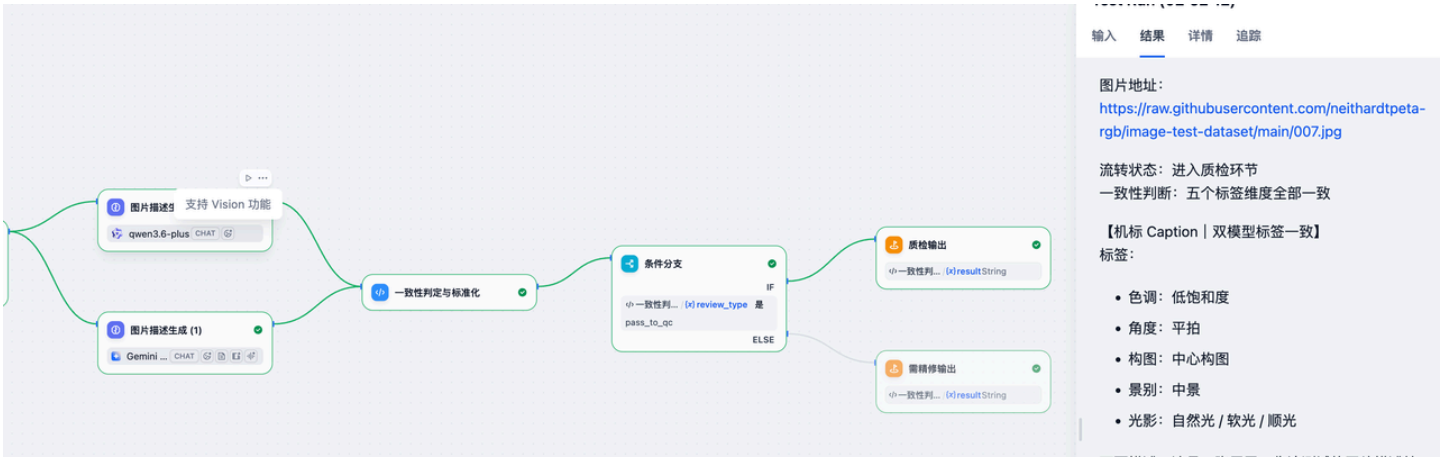


精修 case 2：光影标签不一致，需要人工确认

模型 1 更偏向人工造光、硬光；模型 2 更偏向自然光、软光。这说明光影标签需要人工复核。

2.3.2.1.2 可进入质检 case

当两个模型在 5 个标签维度上全部一致时，结果不会直接视为最终可交付，而是先进入裁判模型复核。裁判模型判断 caption 与图片内容基本匹配、没有明显共同错误时，再进入质检确认环节。



质检 case：5 个标签维度全部一致，可进入质检确认

2.4 最终收益总结

当前能确认的收益主要分为效率收益和质量收益。效率上，标注同学不再需要逐张上传图片、等待加载、复制粘贴 Agent 回复，批量任务可以集中运行；质量上，双 LLM 分流让人工更聚焦分歧样本，裁判模型降低共同错误和低质量 caption 直接流入质检的风险。机标率 47% 需要和具体统计样本范围、统计周期一起使用。

- **图片清洗Work flow**：人效从 17 条/h 提升到 21 条/h 后，每条节约约 40 秒，**人效提升 23.5%**。（上篇引用汇总体现对整个项目交付的加速）
- **Agent 阶段收益**：人效从 21 条/h 提升到 33 条/h 后，每条节约约 62 秒，**人效提升 57.1%**。
- **Work flow阶段收益**：人效从 33 条/h 提升到 46 条/h 后，每条节约约 31 秒，**人效提升 39.4%**。
- **全流程累计**：17 条/h → 46 条/h，**人效累计提升 170.6%**，整体处理能力达到原来的 **2.71 倍**。
- **加入裁判模型收益**：实现部分机标，**机标率 47%**，机标**准确率 91%**。

收益维度	具体变化	项目价值
批量链路打通	从单张 Agent 对话扩展到 CSV 批量运行。	减少逐张上传、等待和复制粘贴的重复动作。
结构更统一	每张图固定输出五维标签、候选 caption 和标准化字段。	降低人工改写和后续解析成本。
复核更有重点	双 LLM 分流识别分歧样本。	人工优先处理高风险样本，而不是平均检查所有样本。
质量兜底更完整	裁判模型复核标签一致样本的 caption 可用性。	降低共同幻觉、主体遗漏和低质量描述进入质检的概率。

可复用性：当前版本已经形成一个可继续扩展的基础链路：CSV 批量输入图片 URL，由 Dify 下载图片，两个视觉模型并行生成标签和候选 caption，再通过一致性判定、裁判模型质检兜底和三类出口完成自动分流，供人工精修、质检和规则复盘使用。

2.5 方法论沉淀与巧思

这次更大的收获，是把人工流程里最重复、最容易拖慢效率的环节拆出来。质量稳定来自双模型一致性判断分流和裁判模型兜底。

- 先把问题拆开：图片清洗解决“能不能标”，图片描述工作流解决“怎么更快、更统一地描述”。
- 先用 Agent 验证描述框架，再用 Workflow 承接批量运行，避免一开始就把流程做得过重。

- 工作流适合先生成可改写底稿，让人工把精力放在调整、精修和异常样本处理上。
- 真正影响效率的往往是上传、等待、复制粘贴、培训使用方式这些重复动作，需要优先被流程化。

2.6 后续上线与风险管控策略

后续优化应从真实使用反馈和 bad case 出发，优先补齐规则边界、分流口径和裁判模型打回样本的复盘机制。

离线测试——仿真测试——小流量测试——全量上线（在每个阶段都使用分层回归测试看调优是否正收益）

1. 根据常见图片类型，整理更贴近业务的描述模板和标签边界。
2. 继续和图片清洗结果接起来，明确只接收合格图片，灰度图片复核后再决定是否进入描述流程。
3. 定期汇总双模型分歧样本和裁判模型打回样本，沉淀为 bad case，用于反向优化标签规则、系统提示词和测试集。

(注：内容由 AI 生成，请谨慎参考)