# Prompt Drift Lab｜v2.0 修改思路与文件级调整路线

> 目标：在**不篡改、不补全任何实验数据**的前提下，将当前仓库统一为一个对外清晰、对内可维护、可复现的“发布版工程”。
>
> 说明：**版本号仅存在于 Git tag / Release 层面**，正文与文件命名中避免反复出现 v2/v3 等内部迭代叙事。

---

## 一、总体设计原则（必须统一）

### 1. 版本号放置规则

- ✅允许：Git tag / GitHub Release 使用 `v2.0`
- ❌避免：正文、README、文件名中反复出现 `v2 / v3 / version2`
- 若需保留演进痕迹：仅在 `CHANGELOG.md` 或 Release Notes 中出现一次

> 读者关心的是“这套方法现在如何使用”，而不是你内部如何从 v1 走到 v2。

---

### 2. 读者路径优先（而非作者写作顺序）

所有文档修改遵循同一条阅读主线：

> **工程入口 → 复现路径 → 结果索引 → 失败归因 → 研究位置与展望**

对应到仓库层级： 1. 项目总 `README.md` 2. `01_实验设计/`（题集、协议、输出结构、威胁与局限） 3. `02_提示词版本/`（Prompt Manifest + 各版本意图差分） 4. `03_评测规则/`（Eval Protocol / Judge Prompt / 有效-无效口径） 5. `04_实验结果/`（结果索引，而不是直接“讲结论”） 6. `05_总结与展望/README.md`

---

### 3. 数据纪律（不可破坏）

- ❌不在 README / 总结中**编造或推导新数值**
- ❌不用“看起来像是提升/下降”的语言替代真实表格
- ✅所有现象都**指向已有产物**：
- `summary.csv`
- `main_method_* / supporting_method_*`
- `judge_*.json`
- 对应 PDF 原始输出

> 文字只起“导航与定位”作用，证据永远在文件里。

---

### 4. 外部资料的使用边界（点缀而非替代）

外部材料只用于： - 帮助读者理解：**你的实验在评测谱系中的位置** - 给导师 / 工业 reviewer 一个“方向对齐”的锚点

明确禁止： - ❌用外部论文结论替你实验“下判断” - ❌用外部 benchmark 的数字类比你当前结果

推荐引用方向（只做定位，不引数据）： - 学术：HELM / IFEval / 指令遵循与可验证评测 - 工业：LLM Evals 工程化、Prompt Injection / Drift 的安全外延

---

## 二、命名与结构统一规则

### 1. 文件名与 README 必须严格一致

- README 中出现的文件名 = 仓库真实存在的文件名
- 历史命名（如 `_v2`）不再出现

### 2. Prompt 版本的叙事方式

不再使用： - “v1 / v2 / v3 Prompt”

统一使用： - **Baseline / Structured / Long / Weak / Conflict** - 每个版本只回答两个问题： 1. 它相对于 Baseline 改动了什么？ 2. 这个改动是为了触发哪一类 Drift 风险？

---

## 三、文件级修改顺序（执行路线）

### Step 1｜项目总 README.md（工程入口）

**目标**：让陌生读者 5 分钟内明白“这是什么 + 如何复现 + 结果在哪看”。

关键动作： - 删除正文中的版本叙事（v2/v3） - 对齐真实目录结构 - 增加“如何读结果”的明确指路

---

### Step 2｜`05_总结与展望/README.md`

**目标**：只做三件事： 1. 概括你实验**已经证明了什么现象**（不引新数据） 2. 说明这些现象在学术 / 工业中的位置 3. 给出**可继续扩展的研究升级方向**（eval 套件化、任务扩展等）

禁止： - 自我否定（“只是初步尝试”） - 编造定量结论

---

### Step 3｜`01_实验设计/`

**目标**：把“复现纪律”写死。

重点文件： - 实验设计_五步法.md：结构化、工程化表达 - 实验协议.yaml：作为“不可随意修改的实验合约” - 标准输出结构.md：评测能否自动化的根基 - 威胁与局限.md：提前替评审回答质疑

---

### Step 4｜`02_提示词版本/`

**目标**：Prompt 不是“灵感集合”，而是**可对照的实验变量**。

- `PROMPT_MANIFEST.md` 中明确：
- 每个 prompt 的设计动机
- 与其他版本的最小差分
- prompt 文件本身不写解释，解释全部集中在 manifest

---

### Step 5｜`03_评测规则/`

**目标**：评测不是“感觉打分”，而是协议执行。

- 明确：什么是 **有效评测**，什么是 **无效评测**
- 把“无效 ≠ 模型失败”写成显式规则
- Judge Prompt 与 Eval Protocol 必须一一对应

---

### Step 6｜`04_实验结果/`

**目标**：给读者一条“从表格到证据”的索引路径。

- 新增或补强一个结果索引 README：
- 哪些 CSV 是总览
- 哪些用于主方法 / 辅助方法
- 如何回溯到具体 PDF + judge JSON

---

## 四、最终状态定义（Done 标准）

当满足以下条件时，可视为 **v2.0 发布态完成**：

- [ ] 正文不再出现版本错位叙事
- [ ] 任一文件名在 README 中都能被准确找到
- [ ] 读者可以不问你，就完成一次复现或结果核查
- [ ] 外部材料只起“定位”作用，不影响你数据的主导地位

---

后续所有修改，都应以本文件为**统一裁判规则**。