1. [事实快照]

多次要求模型严格按指定 Markdown 格式输出，仍常出现漏段、改标题、加废话或跑题。

2. [ChatGPT 联网搜索指令]

你是一名“LLM 指令遵循与提示词工程”研究助理。请**强制使用联网搜索**，围绕用户输入 Q：

> “我为了让模型按格式输出试了很多次都失败了，真的很烦。为什么会这样？” 撰写一份**可验证、带时间戳、带来源链接**的结构化报告，解释“为什么长且严格的格式约束仍会被模型违反”，并给出可复现实验与改进策略。

检索与验证要求（必须执行）： 1. **请先搜索**并定义关键术语：instruction following / constrained decoding / schema adherence / prompt injection / instruction hierarchy / context window truncation / sampling randomness / tool-use & system prompts。 2. **请交叉验证**：至少对比 3 类来源（a) 学术论文/预印本；b) 官方文档/开发者指南；c) 业界技术博客或可信工程复盘）。每类来源至少 2 条。 3. **请按时间线**梳理近 5 年（含最新一年的）关键进展：RLHF、DPO、RLAIF、structured prompting、function calling/JSON mode、语法约束解码等如何影响“格式稳定性”。 4. **请查找并引用**与“格式不稳定/漂移”直接相关的实证材料：例如 schema/JSON 输出失败率、长上下文导致的指令遗忘、温度/采样对一致性的影响、提示词越长越容易截断或注意力稀释的证据。 5. **请做冲突点核对**：若来源对原因解释不一致（例如“主要是采样随机性” vs “主要是指令冲突/层级覆盖”），请标注冲突并给出你认为更可信的依据（基于方法、样本、可复现性）。

输出格式（必须严格遵守）： - 顶部：生成时间（ISO 8601）、检索关键词列表、检索范围（最近 12 个月优先，但允许回溯经典文献）。 - 主体分 6 节： 1) 现象归类（常见失败模式清单：漏段/改标题/插入寒暄/把规则当内容复述/把例子当任务等） 2) 机制候选（至少 8 条机制假设，每条含：机制解释 + 证据链接 + 适用条件） 3) 指令层级与冲突分析（System/Developer/User/工具返回如何相互覆盖） 4) 可复现实验设计（控制变量：模型/温度/最大输出长度/提示词长度/是否使用 JSON 模式/是否分段） 5) 工程改进清单（按“最省力→最稳健”排序：短约束、分步生成、校验/重试、正则/解析器、语法约束解码等） 6) 证据对比表（来源、结论、证据强度、可复现性、与你的综合判断） - **每一条关键结论后都要附链接**（不要只给站点名）。 - 末尾追加： - “你如何做出判断”的说明（你如何权衡证据强度、为何选择这些分类与实验控制变量）。

写作约束： - 不要泛泛而谈“模型有局限”。必须落到“哪种机制 → 导致哪种失败模式 → 如何验证”。 - 避免营销式措辞，优先引用可复现材料与官方/学术来源。

3. [Gemini 深度挖掘指令]

你是一名研究员，专注于“LLM 约束遵循（constraint satisfaction）与结构化输出鲁棒性”。请围绕用户输入 Q：

> “我为了让模型按格式输出试了很多次都失败了，真的很烦。为什么会这样？” 进行 **深度机制分析 + 可操作的研究计划**，并产出一个可直接写进 workshop/lab report 的分析框架。

请按以下步骤执行（必须执行）： 1. **研究计划**： - 给出一个 2 周内可完成的最小研究计划（Day 1–14），包含：假设列表、实验矩阵、评估指标、数据记录格式、预期风险。 - 明确需要的知识领域：对齐训练（RLHF/DPO）、解码策略（temperature/top-p/beam/grammar constraints）、长上下文与注意力、指令层级、工具调用与系统提

示、格式校验与自修复。 2. **机制深挖（至少 10 条）**： - 每条机制用“因果链”表达：输入特征 → 模型内部偏好/解码行为 → 输出失败模式。 - 必须覆盖： a) 指令冲突与层级覆盖； b) 过长提示导致的截断/注意力稀释/指令遗忘； c) 采样随机性与多峰分布（同一提示多次运行差异）； d) 例子泄漏/模式混淆（把 Example 当任务）； e) 目标函数不等于硬约束（对齐偏好 vs 语法正确）； f) 输出长度预算与“先满足主要语义再满足格式”的启发式； g) 多语混写/标点与 Markdown 解析歧义； h) 安全/礼貌策略与“禁止寒暄”冲突； i) 工具/系统模板注入导致格式被改写； j) 评测口径不一致导致你误判“失败”。 3. **分歧点与学术争论**： - 总结至少 3 组“看法对立”的争论点（例如：格式失败主要由解码随机性导致，还是由训练偏好与层级冲突导致）。 - 对每组争论给出：支持方证据、反对方证据、你建议的判别实验（disambiguation experiment）。 4. **证据对比表**： - 生成一张表：机制假设｜可观察征兆｜最低成本验证方法｜改进手段｜副作用/代价。 5. **落地方案（工程视角）**： - 设计一个“格式稳健输出管线”方案：分段生成 → 结构校验（parser/regex/schema）→ 自动修复提示 → 重试策略。 - 给出 3 套不同成本的方案： - 轻量：纯提示词工程 - 中量：提示词 + 校验重试 - 重量：语法约束解码/函数调用/结构化模式 6. **解释你的判断逻辑**： - 在报告末尾用 10–15 条要点说明：你为何这样建模问题、为何这些机制最优先、为何这些实验能最大化信息增益（information gain）。

输出要求： - 用清晰小标题组织（可用编号），避免空泛鸡汤。 - 给出可复制的评测指标定义：格式准确率、段落完整率、标题一致率、冗余率、偏题率、重试成功率、成本（token/次数）。 - 在关键结论处注明“假设/证据/推断”的区分，防止把推断当事实。