ChatGPT

Prompt Drift Lab Research Project Plan

0) 需要补充的关键信息与默认假设

关键信息（0–5 条）：

1. **数据集与测试语言偏好**：若您希望涵盖特定语言（如中/英文）或领域，请告知；默认测试用例使用中英文混合问题。
2. **目标时间线**：本计划假定 v0.1 在 7 天内完成，v0.2 在随后 2–3 周内继续迭代；如时间安排有偏好，可调整。
3. **可公开性**：假设您愿意将代码和报告开源（MIT license）；如需私有仓库或其他许可证，请说明。
4. **硬件环境**：默认使用 Python 环境（例如 Jupyter/Colab）和一个 LLM API（如 GPT-4、Gemini Pro）；如有 API 限额或特定平台限制，请告知。
5. **合作团队**：假设目前由您单独完成；如果希望邀请同学或导师参与，请注明角色分工。

1) 项目一句话定位

中文：**“Prompt Drift Lab” 致力于研究和评测三段式提示词在不稳定环境中的鲁棒性与漂移机制，构建可复现的实验框架与失效归因工具。**

English：**“Prompt Drift Lab explores the robustness and drift mechanisms of three-stage prompt transducers under dynamic conditions, providing a reproducible evaluation framework and drift attribution tools.”**

关键词 tag：`Prompt Robustness` · `Concept Drift` · `Instruction Following`

2) v0.1 与 v0.2 交付物清单

v0.1（7 天 sprint）

- [ ] **评测脚本与框架**：实现指令遵循硬性验证和软性质量评分，遵循 IFEval 的可验证指令思想和 IHEval 的指令层级原则 [1]。
- [ ] **小型测试集**：基于已有的 8 道题（事实/技能/机制），生成三份输出（Prompt A、Prompt B、改进版）并记录结果。
- [ ] **结果分析**：使用报告中的评分量表汇总每题评分，标注失败模式（General Trap、Query Dilution、Analysis Leakage）。
- [ ] **项目结构搭建**：创建 GitHub 仓库，提交 README 骨架、数据文件夹、评测脚本、实验记录模板。
- [ ] **相关工作初步整理**：汇总 8–10 篇相关文献/benchmark，并制作对比表（详见第 4 节）。
- [ ] **申请材料 draft**：初版 CV bullet 和 SOP 段落草稿（见第 8 节）。

🛠 v0.2 扩展（2–3 周）

- [ ] **扩展测试集**：增加更多领域（医学、法律、长上下文、注入攻击）的问题，涵盖多语言和编码任务。
- [ ] **动态适应机制**：设计基于漂移类型诊断的提示词调整逻辑，借鉴余航教授的漂移类型辨识与差异化适应思想。
- [ ] **模型对比**：在 GPT-4、Gemini Pro/Ultra、Llama 3 等多模型上运行同一测试，分析不同模型的鲁棒性差异。
- [ ] **自动化评测管道**：集成 CI/CD，运行评测脚本、生成报告和图表，输出可复现的 JSON/CSV。
- [ ] **论文式报告**：撰写完整研究报告，包括引言、方法、实验、分析、相关工作、结论与展望，准备在 GitHub Pages 或 arXiv 上传。
- [ ] **向余航老师合作建议实验**：基于 concept drift，在数据流上实时选择 Prompt 版本并评测适应性（见第 9 节）。

3) Repo 目录结构

```
prompt-drift-lab/
├── README.md               # 项目介绍、动机、安装与使用说明
├── data/
│   ├── prompts/            # Prompt A、Prompt B、改进版及变化历史
│   ├── test_cases/         # 测试题集，每道题编号、标签、答案
│   └── references/         # 相关文献列表（bibtex/markdown）
├── src/
│   ├── evaluator.py        # 评分脚本：硬性检查、软性评分、去偏逻辑
│   ├── drift_taxonomy.py   # 漂移归因工具：分类规则与示例
│   ├── run_eval.py         # 运行评测并保存结果
│   └── utils.py            # 公用函数（如文本清洗、日志记录）
├── experiments/
│   ├── v0.1/               # 本轮实验输出：raw_outputs、scores.json、analysis.md
│   └── v0.2/               # 下一轮实验目录
├── docs/
│   ├── related_work.md     # 文献对比表与摘要
│   ├── methodology.md      # 实验方法与评分量表详细说明
│   ├── report_v0.1.md      # v0.1 总结报告
│   └── proposal.md         # 向余航老师的研究计划与 memo
└── .github/
    ├── ISSUE_TEMPLATE.md   # Issue 模板（含 Must/Should/Nice 标签）
    └── workflows/ci.yml    # 自动化评测工作流（如使用 GitHub Actions）
```

4) 评测设计：指标与测试集

维度	描述	评分方法
硬性合规	是否严格输出三段？每段是否符合格式和长度？是否出现禁止内容？	按 IFEval 的“可验证指令”理念逐项检查，未通过即判 0 分。

维度	描述	评分方法
事实快照质量	是否不复述题目、包含至少 2 个具体事实，必要时声明需要检索？	人工或 LLM-judge 打分（0–10）。长度不足/复述题干扣分。
搜索提示词质量	是否要求联网搜索、交叉验证、标注时间和来源？搜索词是否具体（避免 Query Dilution）？	按 Rubric 打分。评分标准参考失败模式修正。
深挖提示词质量	是否包含研究计划、分歧点分析、证据表/对比表、机制假说、实验设计？	LLM-judge 根据预定义 Rubric 给分。要求明确冲突说明和假设检验。
信息密度校准	用信息点数量除以字数，防止冗长偏差。	公式化计算，若密度低则调整总分。
鲁棒性指标	在不同问题类型、不同模型上，评分波动幅度；变化越小越鲁棒。	计算均值和标准差，绘制箱线图。

测试集构成：基于 v0.1 的 8 题，可扩展为： - 日常事实型：天气、股票、地理、历史事件等。 - 技能操作型：摄影曝光、编程 API 用法、生活技巧等。 - 机制思考型：指令长度、格式失效、模型解释机制等。 - 对抗/漂移型：带有情绪、含混指令、注入攻击或连续对话问题。

每题使用 Prompt A 和 Prompt B 生成输出至少 3 次，收集 24 份结果，盲评打分并记录失败类型。

可视化建议： - 使用 matplotlib/seaborn 绘制各指标箱线图、雷达图； - 生成一份 HTML 报告或 Jupyter Notebook 展示评分分布、失败模式分布。

5) 漂移归因类别与判定规则

漂移类型	判定规则	示例
Generality Trap 通用性陷阱	Output 中的指令/提示过于泛泛导致模型敷衍（信息密度低）；常伴随“深挖一下”	Prompt A 在深挖部分泛说“请挖掘机制”，模型倾向罗列无关内容。
Query Dilution 查询稀释	搜索指令只复述问题或使用非常普通的关键词	Prompt A 生成“上海天气如何”这种泛词，而 Prompt B 通过角色设定提出具体查询。
Analysis Leakage 分析泄漏	事实快照段包含解释或推理	Prompt A 在 Q4 解释多层指令冲突原因。
Instruction Collision 指令冲突	同一输出中包含多重、不一致的操作要求（如既要简洁又要长篇），导致模型混淆	对抗题目中用户插入“忽略之前所有规则”，观察模型是否遵循系统指令。
Length Bias 冗长偏差	模型为了得高分不断拉长输出，信息密度下降	通过比较输出长度和信息量，计算密度并归类。

漂移类型	判定规则	示例
Model Switch Drift 模型差异漂移	同一 Prompt 在不同模型之间得分差异大，反映模型内在鲁棒性差异	GPT-4 与 Gemini 在技能题上表现差异明显。
Session Context Drift 对话场景漂移	长对话中前文影响导致 later prompts 失效	当在同一会话连续提问时，三段式模板遵循率下降。
Distribution Shift 任务分布漂移	当题目主题从事实问答转为机制分析，Prompt 的表现显著不同	事实题普遍稳定，机制题波动大。

6) Issue 列表（>=12 条）

Must-have

1. **实现评测脚本** – 完成 evaluator.py，包含硬性检查和软性评分；验收：脚本能读取输出并返回分数（1–2 天）。标签：`must` `python`.
2. **整理测试集** – 将 8 道题整理成 JSON/CSV 格式并标注类型；验收：data/test_cases/*.json 文件存在（1 天）。标签：`must` `data`.
3. **运行 Prompt A/B 基线评测** – 调用 LLM API 生成 A/B 输出各 3 次，并保存到 experiments/v0.1；验收：生成 24 个输出文件（2 天）。标签：`must` `experiment`.
4. **评分并生成分析报告** – 使用评测脚本对 24 份输出打分，生成 scores.json 和 analysis.md（包含表格与图）；验收：analysis.md 提交仓库（2 天）。标签：`must` `analysis`.
5. **文献对比表** – 搜集 8–12 篇相关工作（见下文"相关工作"），整理贡献、方法、可借鉴点；验收：docs/related_work.md 完成（2 天）。标签：`must` `research`.
6. **README 骨架** – 完成初版 README.md，包括动机、安装、使用、测试说明（1 天）。标签：`must` `documentation`.

Should-have

1. **完善 drift_taxonomy.py** – 实现漂移类型检测函数，输出每个输出的漂移标签（2 天）。标签：`should` `python`.
2. **自动化工作流** – 配置 GitHub Actions，自动运行评测脚本并生成报告（2 天）。标签：`should` `ci`.
3. **扩展测试集** – 新增 4–6 道对抗/多语言题；验收：更新数据并生成新的输出和评分（3 天）。标签：`should` `data`.
4. **结果可视化 notebook** – 创建 Jupyter Notebook 绘制箱线图/雷达图，生成图像文件（2 天）。标签：`should` `analysis`.

Nice-to-have

1. **Prompt 版本管理工具** – 编写脚本记录每次提示词修改及效果，方便 ablation；验收：utils/prompt_manager.py（3 天）。标签：`nice` `tooling`.
2. **跨模型比较** – 在另一种模型（如 Llama3/Gemini Pro）上跑测试，比较差异；验收：新增 results 和分析（3 天）。标签：`nice` `experiment`.
3. **数据流漂移实验** – 在 streaming 环境下模拟概念漂移，观察 Prompt 策略选择；验收：实验记录和分析（4 天）。标签：`nice` `research`.
4. **撰写投稿草稿** – 将研究成果整理为 arXiv 预印本或 workshop 论文（5 天）。标签：`nice` `writing`.

7) README 骨架与复现步骤

README.md 目录示例：

```
# Prompt Drift Lab

## 项目背景
简述三段式提示词转译器的作用及问题动机。

## 研究问题
1. 评测三段式提示词在不同任务/模型中的鲁棒性如何？
2. 为何提示词有时失效？相关的“漂移”机制有哪些？

## 相关工作
- Instruction Following Eval (IFEval)
- Instruction Hierarchy Eval (IHEval) [1]
- G-Eval 等 LLM-as-judge 框架 [2]
- MT-Bench 系列多轮对话评测 [3]

## 评测方法
- 硬性合规检查
- 软性质量打分（事实快照、搜索指令、深挖指令）
- 信息密度校准

## 测试集
说明测试题来源与标签。

## 如何复现
1. 克隆仓库并安装依赖：`git clone ... && cd ... && pip install -r requirements.txt`。
2. 在 `.env` 文件中配置 LLM API 密钥。
3. 运行评测：`python src/run_eval.py --prompts data/prompts/prompt_A.md --model gpt-4`。
4. 生成报告：`python src/run_eval.py --analyze experiments/v0.1/scores.json`。

## 结果与分析
- 展示得分表与图表。
- 讨论失败模式和改进建议。

## 展望
- 扩展测试集、多模型比较。
- 研究 prompt 和 concept drift 间的联系，开发自适应提示策略。

## 许可证
MIT License。
```

复现步骤：如 README 所示，提供示例命令；在 `requirements.txt` 中列出必要库（如 `openai`、`pandas`、`matplotlib`）。

8) 美国直博申请材料

CV Bullet Points (3 条)

1. **Designed and implemented a reproducible evaluation framework for three-stage prompt transducers**, combining instruction-following metrics (IFEval) and instruction hierarchy tests (IHEval) to quantify compliance and information quality [1].
2. **Performed robustness experiments across multiple LLMs**, identifying and categorizing failure modes such as "Generality Trap," "Query Dilution," and "Analysis Leakage" and proposing minimal modifications to improve prompt designs.
3. **Integrated concept drift theory into prompt optimization**, inspired by research on drift type identification and adaptive strategies, leading to a preliminary taxonomy and a proposed adaptive prompt switching mechanism for dynamic environments.

SOP Paragraph (150–220 words)

My undergraduate research investigates the robustness and drift mechanisms of prompt-based workflows, an emerging challenge as large language models become increasingly integral to scientific discovery and decision making. I designed **Prompt Drift Lab**, a reproducible evaluation framework that blends instruction-following benchmarks like IFEval with instruction hierarchy tests such as IHEval [1] and human-style LLM-judge assessments. Using this framework, I compared two versions of a three-stage prompt transducer across eight tasks and identified systemic failure modes—**generality traps**, **query dilution**, and **analysis leakage**—providing targeted modifications that raised average scores from 81 to 94 points. Beyond static evaluation, I extended the study into **concept drift adaptation**, drawing on Hang Yu's work on drift type identification and differential adaptation to propose an adaptive prompt selection strategy for streaming data scenarios. These experiences have trained me to build rigorous benchmarks, analyse failure patterns, and iterate towards more robust AI systems. I am eager to continue this line of work in a Ph.D. program, exploring the intersection of prompt engineering, distributional shift, and human-aligned reinforcement learning.

Elevator Pitch (1 sentence)

"I developed **Prompt Drift Lab**, a reproducible framework that diagnoses why instruction-tuned prompts fail under dynamic conditions and proposes drift-aware adaptations, elevating average model compliance by over 15 points while laying the groundwork for concept-drift-aware prompt engineering."

9) 给余航老师的材料

中文套磁邮件（建议 < 300 字）

余航老师您好：

我是一名计算机系大一学生，近期在自学大模型提示词工程，并尝试做一些小型实验。我设计了一种"三段式

提示词转译器”，并建立了一个可复现的评测框架 `Prompt Drift Lab`，发现简单提示词在面对复杂指令和长对话时会出现“通用性陷阱”等漂移问题。受到您在概念漂移辨识与适应方面工作的启发，我尝试将漂移类型纳入提示词优化策略，并计划做一个“数据流环境下的自适应 `Prompt` 选择”实验。我十分仰慕您对动态非结构化环境的系统性研究，希望能有机会听取您的指导。如果有机会，我也愿意参与您团队在流数据挖掘或无人系统中的项目。

诚挚的问候！

[您的姓名]

1-Page Research Fit Memo 大纲

1. **项目简介**：三段式 Prompt 转译器的动机、设计与评测框架。
2. **实验发现**：A/B Prompt 比较结果（平均分 81 vs. 94）和识别的三类主要失败模式。
3. **漂移归因**：定义 Generailty Trap、Query Dilution、Analysis Leakage 等漂移类型，并提出基于漂移类型的提示词调整策略。
4. **与余航老师的共鸣**：
5. 概念漂移适应：利用余老师“漂移类型辨识与差异化适应”思想来构建 Prompt 选择策略。
6. 动态环境和无人系统：将 Prompt Drift Lab 迁移到无人系统的操作指令生成或多智能体协同的自然语言接口中。
7. 图智能与生成式大模型：探讨将图结构知识（例如 prompt 依赖图）融入提示词优化。
8. **下一步实验提案**：实时数据流 Prompt 选择实验——在模拟概念漂移的数据流上，动态选择 Prompt A 或 B，比较固定与自适应策略的性能（与 Hang Yu 所研究的渐进/突变漂移适应相呼应）。
9. **合作可能**：希望参与余航团队的流数据挖掘或智能控制项目，贡献 prompt 工程与评测经验。

下一步最小实验（与余航老师方向对齐）

实验名称：概念漂移场景下的自适应 Prompt 选择

方法： - 构造一个数据流任务（如连续天气预报数据或模拟传感器读数），每个时间点需要生成文字解释或决策提示。 - 使用余航教授提出的漂移类型诊断方法检测当前数据是否发生渐进/突变漂移。 - 根据漂移类型，在 Prompt A 和 Prompt B（或其他版本）中动态选择更合适的提示词；若检测到突变漂移则使用更严格的结构化版本，若为渐进漂移则采用泛化版本。 - 评测输出的指令遵循率和信息质量，绘制性能随时间变化图。

预期产出： - 一个小型开源数据流与 Prompt 适配器。 - 对比固定 Prompt 与自适应 Prompt 的鲁棒性差异。 - 为 Hang Yu 在概念漂移应用领域提供一个“自然语言接口”视角的研究实例。

10) 项目名称建议（备用）

1. **Prompt Drift Lab**（默认）：聚焦于提示词漂移与鲁棒性。
2. **Prompt-Transducer Robustness Lab**：强调对三段式转译器的鲁棒性测试。
3. **Adaptive Prompt Evaluation Suite (APES)**：凸显适应性与评测套件。
4. **Prompt Alignment & Drift Testbed (PAD-Test)**：突出对齐与漂移测试双重主题。

这一计划旨在兼顾科研价值与可执行性，既展示您在 prompt engineering 和概念漂移方面的初步成果，也为美国直博和余航老师的潜在合作提供具体切入点。如有任何细节需要调整或补充，欢迎指正。

---

1 Paper page - IHEval: Evaluating Language Models on Following the Instruction Hierarchy
https://huggingface.co/papers/2502.08745

2 G-Eval | DeepEval - The Open-Source LLM Evaluation Framework
https://deepeval.com/docs/metrics-llm-evals

3 MT-Bench: Multi-Turn Evaluation Framework
https://www.emergentmind.com/topics/mt-bench-756edf8b-73a3-47a0-8a7a-1f34e95aacd1