好的，这一回合我先给你「项目新阶段执行计划书」。后续两回合我再分别给 **cursor** 工单 和 **Colab+Gemini** 工单。

项目新阶段执行计划书（给老板/PM）

0. 目标与边界

- 总目标：在 4 周内打通 “数据生成→奖励校准→RLHF 小步试炼→按难度与多样性约束扩量”的闭环，形成可复现的训练与评测流水线，产出一版可对外演示的强化推理能力提升结果（含技术报告与可复现实验包）。参考现有差距分析与阶段性门槛，统一以“先补齐多样性+真实评分，再做影子运行与权重校准，最后才开 **PPO**”为铁律。
- 范围：
  - 本地（Cursor）承担：代码修复、工具链与脚本完善、权重校准与阈值落盘、CI/单测、报告汇总。
  - 云端（Colab+Gemini）承担：并行数据生成与评分、GPU 训练、评测可视化与断点续训支撑。
- 硬性开跑前提：预检 Round2 通过并生成 `reports/preflight/round2_pass.json`；否则禁止启动训练。

---

1. 时间线与里程碑（**4** 周，周一为周期起点）

第 **1** 周：数据与评分体系“打底”

任务

1. 完成 Template Pack v2（每类任务≥6模板；角色≥4、语体≥3），输出 `templates/pack_v2/*.json`。
2. 跑多样性体检：TTR、distinct-1/2、3-gram KL（对 v1）、Zipf 斜率；出 `diversity_report.json`。
3. 打开 **live scoring**，K=3 求 median，记录 variance；高方差样本归档 `unstable_samples.jsonl`。
4. 建立 难度度量与分桶：`difficulty_metrics.py` → `...metrics.jsonl`；`difficulty_bucketize.py` → `rc1_seed.balanced.jsonl`；`validate_difficulty.py` 出直方图/分位数与任务拆分报告。

验收门槛

- distinct-2 ≥ 0.60；与 v1 的 3-gram KL ≥ 0.15；角色≥4/语体≥3；方差>0.08 的样本单列并降权。

- 难度分布：按任务 **Hard ≥30%、Easy ≤30%**；`clue_overlap`（hard 中位数 ≤0.20，easy 中位数 ≥0.40）；长度/轮次/工具链上限不过界；生成 `reports/rc1/difficulty_report.json`。

交付

- `diversity_report.json`、`rc1_seed.metrics.jsonl`、`rc1_seed.balanced.jsonl`、`difficulty_report.json`、`reports/preflight/round1.json`。

第 **2** 周：影子运行与权重校准、惩罚接入

任务

1. **Shadow Run**：旧 7 维 vs 新奖励并行评分（n≈245，分层抽样），输出相关性与 topK 重合。
2. 权重校准：非负最小二乘 + L2 先验（CV=5、bootstrap=200），落盘 `configs/weights.json` 与 `calibration_report.json`；任何单维权重 ≤0.5、权重归一且非负。
3. 过度澄清惩罚（needs_clarification=false 时按回合计罚，α≈0.07，上限 cap=3）接入奖励聚合并做消融。

验收门槛

- Shadow：Spearman ≥ 0.75；Top-10% 重合 ≥ 70%；与“任务成功”相关性的相对提升 ≥ +10%。
- 校准：rank-corr 中位数 +8%↑、MAE -5%↓；`weights.json` 满足非负、归一、单维 ≤0.5。
- 惩罚：过度澄清率 ↓≥20%，平均轮次不升，成功率不劣化>1pp。

交付

- `reports/shadow_run_*.json`、`reports/calibration_report_*.json`、`configs/weights.json`、`reports/overclar_ablation_*.json`。

第 **3** 周：小步 **PPO** 试炼（本地/云端任选一处，建议 **Colab GPU**）

任务

- 5k steps 试炼：起始超参建议 `lr=1e-5, clip=0.2, kl_coef=0.02, batch_size=32, rollout_len=128`；评测 HotpotQA / StrategyQA / GSM8K 成功率、平均轮数、过度澄清率、策略 P/R。
- 训练采样接入 难度加权 与 优先采样（by_difficulty：easy:0.2, medium:0.4, hard:0.4）。

- 建立断点续训：`checkpoints/rc1/step_xxx` 固化；Colab 断线后以 `--resume-from` 恢复。

验收门槛

- “需问类”成功率 绝对提升 ≥ +5pp；过度澄清率 相对下降 ≥ 20%；平均轮次不升。
- “Hard 桶”成功率合并口径（HotpotQA/StrategyQA）提升 ≥ +5pp。

交付

- `logs/train.log`、`reports/ppo_eval_*.json`、`checkpoints/rc1/best/`、`final_model/`（或合规命名）。

第 4 周：扩量与发布候选（**RC1**）

任务

- 按门槛通过后，扩量生成 **1000** → **5000**；每个阶段前后均重跑“多样性+稳定性+难度报告”作为闸门（不过门不扩量）。
- 打通 训练↔推理双轨：训练用 PyTorch/TRL；推理/离线评测支持 `llama.cpp + GGUF`；提供 PyTorch→GGUF 转换占位脚本与 README 说明。
- 产出 RC1 报告与复现实验包（含数据指纹 SHA256、权重与配置摘要、主要曲线与分桶结果）。

验收门槛

- 1000 扩量前后报告均达标后，方可进 5000；最终 RC1 报告包含难度拆分指标与对照曲线。

---

2. 角色分工（**RACI**）

- 负责（**R**）：
  - Cursor（本地）：评测/校准脚本、缓存/限流/惩罚接入、单测与 CI、影子运行与报告聚合。
  - Gemini（Colab）：模板扩写并行生成、live scoring 批评测、GPU 训练与可视化、断点续训。
- 主责（**A**）：你（PM/老板），把控里程碑与闸门，签发“可扩量/可开训”。
- 协作（**C**）：我（架构师），给出工单与门槛、做技术裁决；必要时修订指标与流程。
- 知情（**I**）：仓库协作成员；所有日报/周报默认公开在 repo issues/wiki。

---

3. 度量指标与看板

- 核心 **KPI**：Spearman、Top-10/20 重合、成功率（总/难度分桶/任务分项）、过度澄清率、平均轮数、生成多样性（distinct-2、3-gram KL、角色/语体覆盖）、稳定性（评分方差>阈值占比）。
- 工程 **KPI**：缓存命中率（上限警戒<95% 防“伪真评分”）、429_rate、avg_latency、日志/检查点完备率、数据指纹覆盖率。
- 可视化：TensorBoard/JSON 报告 + issues 图表小结（周末固定产出“周回顾”）。

---

4. 预检闸门（两轮制）

- **Round 1**：种子池固化（多样性/难度/泄漏/长度等全部 PASS）、权重文件存在性与格式校验、磁盘资源 > 50GB。输出 `reports/preflight/round1.json`。
- **Round 2**：在更接近实战口径下复跑 1–5 步，并缓和缓存阈值到 `<95%` 后再做防伪；产出 `round2_pass.json`。只有存在该文件才允许训练。

---

5. 断点续训与合规存档

- 断点续训：所有训练进程按 step 保存 `checkpoints/rc1/step_xxx/`；Colab 端断线后用 `--resume-from` 恢复；每次阶段评测通过后，复制到 `checkpoints/rc1/best/` 冻结。
- 合规存档：对关键产物（影子集、种子池、权重、报告、模型）计算 SHA256，存 `reports/rc1/*.sha256`，并输出 manifest。
- 日志与可审计性：训练/评分/评测日志均保存于 `logs/` 与 `reports/`；评分记录需保留实体打分明细 与 **variance** 字段，便于溯源。

---

6. 风险与应对

- 评分“伪真”风险（过高缓存命中、模拟打分）：设置 live_mode、防伪探针、缓存命中阈值闸门（首轮 <90%，复核 <95%）。
- **Colab** 断线：以 30–60 分钟保存频率落盘检查点到 Drive；训练脚本默认 `--resume-from`；日内多任务分片以缩短不可恢复区间。
- 数据偏难/偏易：使用难度分桶与目标分布控制，并提供快速“提难/降难”旋钮（tool_hops、ops_numeric、clarify_rate、paraphrase 强度）。
- 评测不稳定：对 variance>0.08 的样本降权或重评；并记录 `unstable_samples.jsonl`。

---

7. 本周 To-Do（可落地）

- 批准本计划并冻结 4 周时间线与门槛。
- 我输出两份工单：（1）**cursor** 工单、（2）**Colab+Gemini** 工单，均为“可直接拷贝执行”的脚本化指令与报错上报规范。
- 建立周例会节奏：周三中期检查、周日里程碑验收，凡未过闸门不推进下一步（特别是训练与扩量）。

如果没问题，我下一条消息就发 **cursor** 工单（本地）；再下一条发 **Colab+Gemini** 工单（云端）。