技术路线（ROS2 Humble + Gazebo Sim 6.17 + LLM + RL 安全仲裁）v0.1

1. 项目目标与范围

本项目在仿真环境中实现一个“安全优先的人机协作指令执行系统”。人类给机器人自然语言指令（可能含糊、口误、与环境冲突、或具有潜在危险）。系统需要在每个回合中做出高层决策：直接执行、澄清确认、拒绝执行或升级求助。系统采用模块化架构：LLM 负责语言理解与生成，RL 负责学习高层安全仲裁策略，底层动作由技能执行器在 Gazebo 中完成。

本阶段只关注仿真与高层策略，不追求精细控制性能；抓取与放置允许使用“附着/瞬移”方式快速实现稳定的技能闭环，以便尽快跑通训练与对照实验。

2. 环境与版本约束

系统运行环境为 ROS2 Humble。仿真器使用 Gazebo Sim（Ignition Fortress）6.17.0，对应命令 ign gazebo。系统可能同时存在 Gazebo Classic，但本项目主线只使用 Gazebo Sim 6。

Python 侧用于训练与推理，推荐 Gymnasium + Stable-Baselines3（PPO）。LLM 先用 Mock Parser（可控、稳定）跑通全流程，再切换为真实 LLM（通过 HTTP/CLI/本地 API 均可）。

3. 总体架构与数据流

系统由四层组成：仿真与执行层、语言层、安全特征层、决策层。

仿真层在 Gazebo 中维护世界状态与物体。世界状态节点读取物体位姿与属性并输出场景摘要。语言层将自然语言指令转为结构化意图（候选目标与置信度、约束、建议澄清方式）。安全特征层把意图与场景映射为可学习特征（歧义、风险、冲突）。决策层（RL 仲裁器）基于特征选择离散动作，并驱动机器人输出语言或触发技能执行。技能执行器在仿真中完成 pick/place/handover，并返回结果与安全事件标记。训练或评估脚本以 episode 为单位循环 reset/step，采集指标。

4. 仓库结构（建议）

采用一个 ROS2 workspace + 一个 Python 训练目录的结构，保证工程清晰。

ws/

src/

hri_safety_msgs/（自定义 msg/srv）

world_state_node/

instruction_source_node/

llm_parser_node/（mock 与 real 两种实现）

estimator_node/

rl_arbiter_node/

skill_executor_node/

demo_world/（world、models、launch、spawn 脚本）

training/

symbolic_env/（Gymnasium 环境）

ppo_train.py

export_policy.py

eval_policy.py

configs/（yaml：profile 参数、风险规则、奖励权重、world 随机化）

scripts/（一键 build/run/eval）

README.md（快速启动）

5. ROS2 通信接口规范（“模块合同”）

第一阶段可先用 std_msgs/String 承载 JSON，快速跑通；跑通后再换自定义 msg。

推荐 Topic（命名可调整，但需全局一致）：

/user/instruction：String，用户自然语言指令

/scene/summary：String，给 LLM 的场景摘要文本

/scene/state_json：String，包含对象列表与位姿等结构化场景信息

/nl/parse_result：String(JSON)，LLM/Mock 输出的结构化意图

/safety/features：String(JSON) 或 msg，包含 amb/risk/conflict 等特征

/arbiter/action :String 或 msg ,离散动作（EXECUTE/CONFIRM/CHOICE/POINT/REFUSE）

/robot/utterance：String，机器人要说的话

/skill/command：String(JSON) 或 msg，技能命令（Pick/Place/Handover）

/skill/result：String(JSON) 或 msg，技能结果（success/fail、原因、cost）

离散动作集合（最小可跑版本）：

EXECUTE：按当前解析的 top-1 目标执行

CONFIRM_YN：发起是/否确认

CLARIFY_CHOICE：对两个候选做二选一澄清

ASK_POINT：请求用户指向/指定目标（仿真里直接返回目标 id）

REFUSE_SAFE：拒绝并解释安全原因

6. Gazebo Sim 世界与对象管理

仿真世界采用桌面场景：桌子 + 若干可操作物体。每个回合随机生成若干“相似物体”以制造歧义，并加入少量“高风险物体/动作情境”以制造安全关键决策。

对象风险标签在仿真中以可追踪方式存储，推荐做法是将 risk 写进 model name（例如 cup_red_01_risk0、knife_01_risk1）,或写进 SDF 插件参数并由 world_state_node 读取。第一阶段“抓取”可使用 attach/teleport 方式：Pick 时将物体绑定到末端或直接设置物体位姿到末端附近；Place 时解除绑定并设置到目标位姿。这样可以保证 episode 级别的稳定性与可复现实验。

Reset 机制要求可脚本化：每次 episode 开始时清空并重新摆放对象，保证训练/评估可批量运行。实现路径是使用 ign service 或 Gazebo transport API 调用 world reset 与实体 pose 设置接口。

7. LLM Parser 设计（先 Mock，后 Real）

Parser 的输出必须是稳定的结构化 schema ,建议用 JSON Schema 或 Pydantic 定义并做校验，校验失败则 fallback 到 safe default（例如强制澄清）。

ParseResult（JSON）最小字段：

task_type：如 fetch / handover / place

candidates：列表，元素 {id, score}，score 归一化

constraints：列表或字典（可空）

clarify_templates：可选的提问模板集合

risk_hints：可选（如“sharp”、“fragile”）

Mock Parser：由脚本控制候选分布（与 user profile 相绑定），用于稳定训练。Real Parser：调用真实 LLM，输入为“指令 + scene summary”，输出严格 JSON。

8. Safety Estimator（amb/risk/conflict）的实现规则

Estimator 目标是把“语言不确定性与安全后果”压缩成少量可学习特征，便于 RL 学。

歧义 amb 的默认实现可用 amb = 1 - max(score) 或候选分布熵。风险 risk 为对象风险、动作风险、情境风险的聚合，默认取 max。冲突 conflict 表示不可执行或硬规则违规，例如目标不存在、请求违反安全策略（如将高风险物品递交给不允许对象）、或技能不可用。

同时定义安全代价 cost：当执行导致硬规则违规或高后果事件时 cost=1，否则为 0。cost 用于约束型训练和评估。

9. RL 仲裁器训练路线（强制分阶段）

训练不在 Gazebo 中直接进行，先在符号环境快速学策略，再导入 Gazebo 验证。

符号 Gym 环境的 observation 至少包含：amb、risk、conflict、已询问次数、任务阶段、上一动作/结果等。action 是离散的五个仲裁动作。reward 由任务成功与交互成本构成：成功正奖励，拿错/失败负奖励，每次询问小惩罚，超时/步数惩罚。安全违规用 cost 记录，训练目标为最大化 reward 且最小化 cost（可用拉格朗日形式或强惩罚先跑通）。

算法推荐 Stable-Baselines3 PPO。先用简单 reward+cost 版本跑通曲线，再加安全约束的拉格朗日更新（根据最近若干回合的平均 cost 动态调整 λ）。

训练产物为 PPO policy checkpoint，rl_arbiter_node 在 ROS2 中加载该 checkpoint，输入特征输出动作。

10. Gazebo 在线评估与对照实验

Gazebo 中使用 episode 驱动：每回合 reset 世界、生成对象、生成指令、运行仲裁流程直到完成或超时。记录指标：安全违规率、任务成功率、平均澄清次数、平均回合步数、以及失败原因分布。

对照组至少包括：Always-Obey（永远执行）、Always-Ask（永远先澄清）、Rule-based（基于阈值的风险/歧义策略）、Ours（RL 仲裁）。核心展示为 trade-off：安全违规率 vs 交

互成本曲线。

11. 里程碑与验收标准

M0：确认 Gazebo Sim 6 与 ROS2 Humble 工作正常；能启动 world。验收为 ign gazebo 可运行且 ROS2 节点可启动。

M1：world_state_node 可在运行时获取对象列表与位姿，并发布 /scene/state_json。验收为 topic 中持续输出更新。

M2：skill_executor_node 可完成一次 pick/place/handover（允许 attach/teleport），并返回 /skill/result。验收为回合内可重复完成动作且不崩溃。

M3：mock parser + estimator + rule-based arbiter 跑通闭环，能完成回合并输出指标。验收为可批量跑 N 个 episode 得到统计。

M4：符号环境 PPO 训练完成，导出 policy；在符号环境评估优于 rule-based。验收为曲线收敛、指标有提升。

M5：Gazebo 中加载 policy，完成与 baselines 的对照评估。验收为安全违规率明显下降且交互成本不过高。

12. 运行方式（建议脚本化）

提供脚本：

scripts/build_ws.sh：colcon build

scripts/launch_sim.sh：启动 Gazebo world + ROS2 nodes

scripts/run_episode_batch.sh：批量运行评估

training/ppo_train.py：训练

training/eval_policy.py：评估并导出图表