模型训练与迭代规划：提升评论回复准确度 & 客户定位/预测

本文面向当前仓库（评论区智能回复 + 知识库检索 + 潜客识别 + 运营监控），给出一套可落地的“先把系统跑稳、再用数据把模型训准”的路线图：你应该训练什么、需要哪些数据、怎么评测、以及如何在现有代码结构中接入。

0. 结论：现在是否“必须训练模型”？

不建议一上来就训练生成模型（大模型微调），原因是**缺少高质量的监督信号/评测集时，训练会把问题放大**：训练数据偏、标注不一致、知识库不完备都会让模型“更自信地胡说”。

更稳的顺序是：

1. **先补齐数据闭环与评测**（让“准不准”可度量、可回放、可迭代）
2. **先训练小模型/轻量模块**（意图识别、潜客预测、检索排序）来获得稳定收益
3. 在有了**足量高质量“好回复”样本 + 业务反馈**后，再考虑**微调生成模型**或引入更强的基座模型

这条路线在你当前项目里尤其合适：仓库已经具备 RAG（知识库/向量检索）、规则意图与潜客评分、监控日志，但**还缺“训练与评测所需的标签/反馈”**。

1. 现状速览（对齐当前仓库能力）

当前后端链路大致是：

- 意图识别：规则（buy_intent / after_sales / complaint / question / praise / chat / empty）
- 潜客评分：规则加权（0-100，low/medium/high）+ 建议动作
- 检索：Embedding + 向量召回 + 词面相似度混合打分（TopK）
- 生成：有 GLM Key 走 LLM；无 Key 走模板兜底
- 合规：脱敏（手机号/微信/外链）+ 长度控制
- 监控：记录回复事件（意图、潜客分、耗时、是否用 LLM）

示例数据主要来自 `xhs/json/search_contents_*.json` 与 `xhs/json/search_comments_*.json`。

2. 把目标说清楚：你要“变准”的是什么？

建议把“准确度、客户定位、预测”拆成 4 个可评测的子目标：

2.1 回复质量（Reply Quality）

你真正需要提升的是：

- **事实一致性（Faithfulness）**：回复不能编造，不与知识库冲突
- **相关性（Relevance）**：对准评论问题，少废话
- **可执行性（Actionability）**：给出下一步（追问关键信息 / 推荐对比点 / 售后路径）
- **风格合规（Policy）**：不输出外链/联系方式/隐私信息；不硬广；语气自然

2.2 客户定位（Customer Profiling / Segmentation）

“定位”更像是**结构化画像**，例如（可按你业务裁剪）：

- 购买阶段：浏览 / 种草 / 对比 / 决策 / 已购 / 售后
- 关注点：价格/活动、功效、成分、安全、适配、发货、售后
- 约束：预算区间、肤质/敏感、使用场景、型号/尺码
- 情绪：正向 / 中性 / 负向（投诉风险）

2.3 预测（Prediction）

“预测”需要明确预测目标（label），典型有：

- 购买意向强度（替代/增强现有潜客分）
- 成交概率（需要业务闭环：是否转化/下单/私信）
- 退款/投诉风险（售后/负评倾向）

没有真实业务结果时，也可以从“代理标签”起步（例如：评论里强购买词、是否继续追问、是否出现售后词），但要明确它只代表“短期语言信号”。

2.4 检索质量（Retrieval Quality）

RAG 系统里，很多“回复不准”其实是**检索不准**：

- 没召回正确知识（Recall 低）
- 召回了但排序不对（Rank 差）
- 知识条目本身不完整或写法不利于检索

因此检索也必须进入你的评测与训练范围。

3. 你需要哪些东西（数据、标签、工具、资源）

3.1 最核心的数据表（建议统一成训练样本格式）

建议把每一次“生成回复”的样本，统一成下面结构（不要求一次到位，但越早统一越好）：

```
{
  "sample_id": "uuid",
  "platform": "xhs",
  "note": {
    "note_id": "string",
    "title": "string",
    "desc": "string",
    "tags": ["string"]
  },
  "comment": {
    "comment_id": "string",
    "text": "string",
    "created_at_ms": 1766568292000
  },
  "user": {
    "user_id": "string",
    "nickname": "string"
  },
```

```
  "retrieval": {
    "kb_id": "uuid",
    "kb_version": 1,
    "query": "string",
    "hits": [
      {"chunk_id": "uuid", "content": "string", "score": 0.42}
    ]
  },
  "model": {
    "generator": "glm|template|other",
    "prompt_version": "v1",
    "temperature": 0.3
  },
  "outputs": {
    "reply_text": "string",
    "intent_pred": "buy_intent",
    "lead_score_pred": 85,
    "lead_level_pred": "high",
    "profile_pred": {
      "stage": "comparison",
      "concerns": ["price", "skin_sensitivity"],
      "risk": "low"
    }
  },
  "labels": {
    "intent_true": "buy_intent",
    "lead_level_true": "high",
    "reply_quality": 4,
    "reply_is_faithful": true,
    "reply_is_compliant": true,
    "best_reply": "string",
    "conversion_7d": null
  }
}
```

其中 `labels` 可以先空着，通过人工/运营面板逐步补齐。

3.2 你已经有的数据（可直接利用）

- XHS 示例语料：帖子/评论 JSON（可作为预训练/弱监督/提示词评测样本池）
- 监控事件：每次生成的意图、潜客分、耗时、LLM 是否启用
- 向量检索日志：query、top_k、latency（建议后续补齐"命中的 chunk 列表"）

3.3 你缺的关键数据（决定能不能训练"预测模型"）

要把"客户预测"训准，**必须有结果标签**，至少满足其一：

- 成交/下单（是否、金额、品类、时间窗）
- 私信/加购/点击（行为日志）
- 售后/退款/投诉（风险标签）

如果暂时拿不到真实结果，建议先做两件事：

1. 用"代理标签"把模型跑起来（例如：强购买词 + 追问次数 + 情绪），先验证链路
2. 从产品/运营侧推动最小闭环：至少能记录"是否产生后续互动/是否转私信/是否解决问题"

3.4 标注与反馈工具（建议）

最小可用做法（无需引入复杂平台）：

- 在前端回复面板加 3 个按钮：
  - "这条回复好/一般/差"
  - "意图分类对/不对（可选正确类别）"
  - "潜客分高/中/低是否合理"
- 可选增加：编辑框让运营给出"更好的回复"（作为 SFT 样本）

这些反馈只要进入数据库，就能形成训练集与回归测试集。

4. 评测体系（不评测就无法迭代）

建议同时做离线评测与在线监控。

4.1 离线评测集（Offline Benchmark）

抽样构建一个固定评测集（例如每个意图 50-200 条）：

- 覆盖不同产品/话题/场景（咨询、价格、成分、过敏、售后、负评）
- 覆盖不同长度与噪声（表情、错别字、隐晦表达）
- 保持版本稳定：每次修改 prompt / 检索策略 / 模型都要跑一遍

4.2 指标（Metrics）

回复质量（建议人工为主，模型为辅）：

- 有用性评分（1-5）
- 事实一致性（是否与知识库冲突）
- 合规率（是否含联系方式/外链/隐私信息）
- 追问质量（是否追问"关键参数"而不是泛泛而谈）

意图/潜客预测：

- 意图分类：Accuracy / Macro-F1（类别不均衡时更重要）
- 潜客预测：AUC（若有成交标签）/ 分层准确率（high/medium/low）

检索：

- Recall@K：正确知识是否被召回
- nDCG@K：正确知识排序是否靠前

4.3 在线监控（Online）

在监控里增加维度（至少做版本打点）：

- prompt_version / retrieval_version / model_version
- 是否使用了画像/预测

- 运营反馈（好/差）占比

这样才能做 A/B 或灰度。

5. 模型路线图（从轻到重，收益最大化）

Phase A：不训练也能显著提升（建议先做）

1. **知识库质量提升**
   - 把"容易被问到的问题"写成短条目（Q/A 或要点式）
   - 给知识条目加结构：适用场景、禁忌、对比点、售后路径
2. **RAG 检索增强**
   - Query 改写：将意图 + 关键槽位（预算/肤质/型号/场景）拼入 query
   - 多路召回：向量 + 词面（当前已做）+ 规则关键词召回（可选）
   - 结果重排：对 TopK 进行 rerank（可先用启发式，后续再训练）
3. **Prompt 工程与约束**
   - 明确要求"引用知识要点，不要编造"
   - 缺信息时先追问（尤其是适配类问题：肤质/预算/使用场景）
   - 高意向用户：更明确的下一步（但避免硬广）

这阶段的特点：**不用训练，但需要评测与闭环**，往往能拿到最稳定的 30%-60% 改善。

Phase B：训练轻量模型（最推荐的"第一批训练"）

这一阶段的目标是让"分类与预测更准、更一致"，并把生成模型的负担降下来。

B1) 意图识别模型（替换/增强规则）

- 输入：comment_text（可拼 note_title/note_desc）
- 输出：intent（多分类）
- 训练数据来源：
  - 运营对意图的纠正
  - 规则弱标注（先自动打标签，再人工抽查修正）
- 模型选择：
  - 先用轻量：TF-IDF + 线性分类器（CPU 即可）
  - 再升级：中文小型 Transformer 分类器（需要少量 GPU）

B2) 潜客/成交预测模型（增强现有 score_lead）

- 输入特征：
  - 文本特征（同上）
  - 规则特征（当前已有的 buy_strong/buy_weak/... 可以直接当特征）
  - 互动特征（是否追问、是否多轮、是否出现私信/加购等行为）
- 目标标签：
  - 有业务闭环：成交/私信/加购（强烈建议）
  - 无闭环：代理标签（仅做过渡）
- 输出：
  - 分数（0-1 概率）或 high/medium/low

B3) 检索 rerank 模型（小成本大收益）

很多“回复不准”是因为 TopK 里正确知识排不到前面。

- 训练样本：(query, chunk, label) 其中 label 表示“这条知识是否对当前问题有用”
- 模型选择：
  - 先用 BM25/词面规则做 baseline
  - 再做 cross-encoder rerank（效果更好，但计算更贵）

Phase C：微调生成模型（最后做，收益上限高但风险也高）

当你满足以下条件时再进入本阶段：

- 有稳定评测集 + 可回归
- 有足量“好回复”样本（运营确认、最好包含“更优回复”）
- 知识库相对完备，且检索链路稳定

C1) 监督微调（SFT / LoRA）

训练样本格式建议包含检索到的知识（让模型学会“用知识而不是臆测”）：

```
{
  "input": {
    "note_title": "...",
    "note_desc": "...",
    "comment_text": "...",
    "intent": "question",
    "knowledge": ["...要点1", "...要点2"]
  },
  "output": {
    "reply": "..."
  }
}
```

关键点：

- 训练集中必须包含“不知道就追问”的样本，否则模型会倾向乱答
- 训练集中必须包含“合规负例”（包含手机号/外链的错误输出），并让模型学会拒绝

C2) 偏好优化（DPO / RLAIF）

如果你能收集到“多候选回复 + 运营选择最佳”，就可以做偏好训练：

- (context, reply_good, reply_bad)
- 目标：让模型更稳定地产生“运营喜欢”的风格与策略

6. 客户画像（定位）怎么做：结构化抽取优先于“让模型自己猜”

推荐先做“结构化抽取”（可用 LLM 或轻量模型），再把画像喂给生成与预测模型。

6.1 画像字段建议（可裁剪）

- stage：browse / consider / comparison / ready_to_buy / purchased / after_sales
- concerns：price / efficacy / ingredient / safety / skin_sensitivity / shipping / warranty / authenticity
- constraints：budget_range、skin_type、region、model_size、use_scene
- sentiment：positive / neutral / negative
- risk：complaint_risk（low/medium/high）

6.2 抽取输出建议用 JSON Schema（减少漂移）

让抽取模型输出严格 JSON（缺失字段填 null，不要臆测）：

```
{
  "stage": "comparison",
  "concerns": ["price", "skin_sensitivity"],
  "constraints": {
    "budget_range": null,
    "skin_type": "sensitive",
    "region": null
  },
  "sentiment": "neutral",
  "risk": "low"
}
```

这类结构化输出的好处是：

- 可直接进入数据库，便于统计与回放
- 可作为预测模型特征
- 可作为生成 prompt 的“约束条件”

7. 在当前代码结构中的落点（最少改动路径）

你现在的后端模块边界很清晰，建议按“低侵入”方式接入：

7.1 输入与输出结构（接口层）

- 在 `reply` 的请求结构里扩展可选字段：
  - `customer_profile`（画像 JSON）
  - `predict` 开关（是否启用预测）
- 在响应中返回：
  - `profile_pred`（若启用）
  - `model_versions`（便于监控与回放）

7.2 回复链路（组装层）

- 在构造检索 query 时拼入画像摘要（提升召回）
- 在 prompt 中加入画像与预测结论（提升相关性与策略一致性）
- 保持合规过滤为最后一道硬闸（不可移除）

7.3 潜客链路（预测层）

把现有规则打分改造成"特征工程的一部分"：

- rule features + text features → 预测模型输出
- 同时保留规则解释（signals/next_actions）用于可解释性

7.4 监控与数据闭环（训练数据来源）

建议把以下内容写入监控 meta：

- prompt_version / retrieval_version / model_version
- used_knowledge 的 chunk_id 列表（用于检索评测）
- 运营反馈（好/差、修正意图、修正分层）

8. 资源清单（你需要准备什么）

8.1 人与流程

- 业务 owner：定义"什么是好回复""什么是高意向"
- 标注人员（可兼职运营）：每周稳定标注一小批，形成累积
- 工程：把反馈入口与日志落盘打通

8.2 数据量建议（经验值）

- 意图分类：每类 200-1000 条即可起效（类别越多越需要更多）
- 潜客/成交预测：至少 1000+ 且要有结果标签（否则模型不稳定）
- 生成微调：高质量样本 2k-20k（越干净越重要）
- rerank：标注 (query, chunk) 级别样本 5k-50k（可渐进）

8.3 算力与环境

- 轻量分类（线性模型）：CPU 即可
- Transformer 分类/rerank/LoRA：建议单卡 GPU 起步
- 如果你不想自建训练环境：也可以走第三方平台，但要考虑数据合规

9. 风险与合规（必须提前定规则）

- 不采集/不存储：手机号、微信号、身份证、地址等敏感信息（或入库前统一脱敏）
- 训练数据要可追溯：来源、版本、标注人、标注规范
- 生成侧必须保留硬性合规过滤（再好的模型也会偶发越界）
- 评测必须覆盖"负样本"：投诉、辱骂、引战、诱导留联系方式等

10. 最小可执行清单（按优先级）

1. 固化离线评测集（从现有 XHS JSON 抽样 + 人工标注）
2. 在前端加入运营反馈入口（好/差 + 纠正意图/潜客分层 + 更优回复）
3. 在数据库里落盘反馈（让它能自动汇总成训练集 JSONL）
4. 先训练意图分类器与潜客预测（替换/增强规则）
5. 做检索 rerank（先启发式，后训练）

6. 最后再做生成微调（SFT/偏好训练）