接单宝经营模型收口报告

接单宝经营模型收口报告

生成时间: 2026-11-16 范围: 门店布局可视化 + 参数消费审计 + 12 个月经营剧本模拟器 状态: 全部完成并回归通过

一, 结论先行

本轮把「算得出但用不上」的经营模型补成了可操作, 可视化, 可验证的决策工具链.

· 门店布局编辑器: 拖家具 → 实时重算指标 → 财务联动 → 保存加载, 内网 8080 正式发布, 附 3 套可对比方案库. · 参数消费审计: 四路探针实测每个参数是否真的驱动经营结果, **59/74** 已接线, 真缺口 **0**; 过程中挖出并修掉 **3** 个真 **bug**. · **12** 个月经营剧本模拟器: 6 个场景 × 12 月逐月轨迹, **21** 条不变量断言全绿, 数值全部复用现成内核, 未新写任何公式.

唯一阻塞: 门店真实经营基线仍缺失. 所有经营结果一律标注为「模型推导值」, 不得包装成真实经营数据.

二, 参数消费审计: **59/74** 已接线, 真缺口 **0**

2.1 审计方法本身先修了

原审计脚本只扫 dynamic_model.py + wargame_engine.py, 漏掉门店三大表, 布局模型, 演化链三路消费面, 把 **23** 个实际已被消费的参数误报成「提到但无效果」.

修成四路探针矩阵:

探针	覆盖	关键点
单技师模型	DM.compute(v, P)	主输出 diff
门店模型	FIN.derive(conn, P)	必须拉平三大表行值, 只 diff drivers 看不到只进报表的参数
布局模型	LM.compute_metrics(layout, P=…)	只探源码里 pv(P, …) 实际读的 id 白名单 (盲探 74 个会超时)
演化链	DM.evolve(P, months=12, overrides=…)	硬编码在演化里的参数只在这里显形

2.2 挖出的 **3** 个真 **bug**

编号	现象	根因	修法
1	S16 门店初始投入 20 万只出现在回本周期的说明文案里	值从未进入任何计算值	接线为资产负债 BS-18 / BS-19 备查行, 不动既有任何财务行, 基准输出零变化

编号	现象	根因	修法
2	I23 设备一次性投入参数永远无效	dynamic_model.evolve 里硬编码 **300**	改为读参数 v.get("I23", 300), 基准行为等价但参数真正生效
3	审计漏扫三路消费面	扫描范围只写了两 文件	扩到全消费模块 + 补齐探针矩阵, 误报 23 → 0

2.3 剩余未消费参数**:** 全部给出语义归类**,** 不硬塞

真缺口 = 无消费效果 且 无合理解释. 其余一律按语义归类:

· 对照线 / 判断线: I17 失业金线,I21 机会成本线,I22 门店单均贡献线 · 开关/ 护栏: L06 / L07 / L08 · 合规资质域: R08 / R09 / R10 / R11 · 派生量 (交叉校验用): S09 月获客成本 = S11 × S10 · 数据缺口 (base=0, 等实测): M06 目标客群规模,M07 渗透率 · 产业链参考值 (非计算源): C03 / C05

硬塞这些参数 = 重复计价.S09 若再接一次, 就等于把 CAC 算两遍.

三**,** 门店布局可视化与交互线

3.1 计算内核 **(**纯 **Python,** 与渲染层解耦**)**

文件	职责
circulation.py	通道宽度加权路径搜索 (单源共享 Dijkstra)
layout_model.py	布局指标: 工位产能 / 等候折损 / 坪效 / 动线/ 净距
layout_server.py	8090 HTTP API: /metrics /layout /types /save /custom + 财务联动
static/layout_editor.html	Canvas 拖放编辑器 (只画图收事件, 不算数值)

铁律: 指标只在 Python 计算, 浏览器一律 POST /metrics 取回数值. 前端自己算 = 双份口径必然漂移.

3.2 性能**: /metrics 24s → 2.8s**

原实现对每个目标的每个可进入邻格各跑一次 A* (12 目标 × 上百邻格= 上千次搜索), 实测 24 秒, 拖放直接不可用.

改成一次宽度加权 **Dijkstra** 算出入口到所有格的最优路径, 各目标只做回溯 —— A* 语义上只是 Dijkstra + 启发式, 结果完全等价. 实测 24s → 2.8s.

3.3 六个真 **bug (**算法层**,** 全部代码实证**)**

1. 自由跨度算错: 用了 min(前向, 后向) → 8.8m 走廊被算成 0.15m. 正确公式 = 前向 **+** 后向 **− 1**
2. 通道宽度用 2 × clearance: clearance 只反映贴近某物那一侧, 单侧贴物 + 另一侧开阔会被误判成 0.1m 窄缝
3. 纯网格测宽出分辨率伪影 (报 0.01m 假窄点): 宽度必须用解析射线法量真实矩形, 网格只管找路
4. 栅格化用 floor/ceil: 浮点误差多吃半格, 0.2m 缝被量成 0.10m. 改格中心判定
5. A* 会钻家具缝: 椅子间 0.2m 缝在网格上「可走」但人不会走. 加净宽**< 0.30m** 硬拒 (罚分会被绕路抵消, 必须硬拒)
6. 物件净距用格心 clearance 得 0.05m 假值: 改矩形对矩形解析间隙 (_rect_gap)

3.4 布局 → 财务接线 **(**口径已实测**)**

· 门店级: financials.derive(conn, P, layout_orders=…, layout_techs=…) · 技师级: dynamic_model.compute(v, P, layout={"station_cap_month": X, "techs": N}) · 新客数必须随布局单量等比收缩 (S10 × layout_orders / S02): 产能惩罚已经过单量传导, 新客不缩就是 CAC 摊在更少单量上 = 惩罚算两次

验证脚本 verify_layout_finance.py 三条链 **11/11 PASS**.

3.5 产能上限是「有效工位 **×** 人均单量」

有效工位 = min(工位数, 技师数). 技师数是更硬的约束 —— 加第 6 个工位不提升任何产能 (技师还是 5 个), 坪效分母也是有效工位. 所以「多摆一个位」类方案要先问: 多出来的位有人用吗?

3.6 方案库 **(**让「对比」有东西可比**)**

方案	服务位	技师	单量	收入	EBIT	说明
A_5 工位标准	5	5	480	¥95,040	−¥10,079	基准
B_4 工位紧凑	4	5 (闲置 1)	384	¥76,032	**−¥11,523**	少一位反而更亏
C_4 工位加候北	4	5 (闲置 1)	384	¥76,032	−¥11,523	等候位 6→10, 单量不变

> B 与 C 的对比说明: 瓶颈在工位不在等候区. 面积给等候区不解决产能问题. C 方案若想提单量, 该加的是第 5 个工位并配第 5 个技师 (S01), 而不是第 6 个工位.

四,12 个月经营剧本模拟器

4.1 架构 **(**复用现有内核**,** 未新写公式**)**

```
scenarios.py        场景定义（声明式）+ 编排器
run_scenarios.py    CLI: --scenario <name> | --all → CSV + 中文摘要
verify_scenarios.py  21 条不变量断言
```

· 门店级口径 → financials.derive (内核 = store_breakeven.calc_profile('store')) · 技师级口径 → dynamic_model.compute · 同一行并存两套口径时列名带 store_ / tech_ 前缀, 只做同口径比较 · 参数覆盖全部落在 params 表 [lo, hi] 区间内 (断言校验), 不臆造系数 · 场景假设 (爬坡系数 / 旺季窗口 / 竞品进入月) 写在 assumptions 字段并逐条打印

4.2 六个场景的实测结论

场景	关键结论
基准稳态	12 个月零扰动 (回归锚点), 保本 1150 单, 缺口 670 单
基准爬坡	月净现金流第 **10** 月转正, 现金最低点 ¥35,274, 12 个月未转负
旺季	6-9 月单量 480→624, EBIT −¥7,913 (减亏 22%)
竞品进入	净利率 −10.6% → **−35.5%**, 第 **8** 月现金转负, 期末 −¥60,463
技师流失	单量 480→288, 净利率 −22.7%, 12 个月现金未转负
平台抽成 10%→20%	收入一分不变, EBIT −¥19,583, 第 **11** 月现金转负

4.3 两个必须记住的结论

□ 竞品进入后没有有限保本点. 单均边际贡献转为 −¥17.0/单 ≤ 0 ——不是「多跑几个月能回本」, 是每做一单都亏一单. 这种情况下唯一正确动作是缩量或退出, 不是拉新.

□ 平台抽成上调打的是收入侧. 抽成从 10% 涨到 20%, 营业收入完全不变 (¥95,040), 但 EBIT 从 −¥10,079 直接掉到 −¥19,583 (Δ−9,504 = 收入 × Δ 抽成率), 第 11 个月现金烧穿. 平台规则风险比需求下滑更致命, 因为它砍的是每一单已经做出来的生意.

4.4 断言覆盖 (21 条)

基准零扰动 (首月与现有模型逐值相等) / 产能降幅 ≥ 人数降幅 / 收入口径自洽 (收入 = 单量 × 客单价) / 提成 = S03 × 单量 (人力成本不重复计价) / 全轨迹无 NaN/None (2304 个数值单元) / inf 只出现在语义明确处 (跑道 / 无解保本) / LTV-CAC<3 的行全部标记 cac_stop / 场景声明的期望值全部兑现 / 参数覆盖全在区间内 / CSV 落盘行数正确.

五, 可视化与交付

入口	地址	内容
内网工具页	http://192.168.9.101:8080/jdb_tools/	全套工具导航
布局编辑器 (8080 托管)	http://192.168.9.101:8080/jdb_layout_editor/	拖放 + 实时指标 + 财务联动 + 保存加载
剧本看板 (8080 托管)	http://192.168.9.101:8080/jdb_tools/scenario_board.html	6 场景 × 12 月三线图 + 结论卡
直连后端	192.168.9.101:8090 / 8091	8090 布局计算, 8091 剧本推演

工程化要点:

· 裸端口服务配自愈守护脚本 (socket 探活 + 60s 轮询), 机器重启不哑 · 8080 托管静态页 + 跨源调后端: 服务端需 CORS 头 + 处理 OPTIONS 预检 · 页面加 no-cache meta (否则手机拿到旧版) · HTTP 响应写盘包 try/except (BrokenPipeError, ConnectionResetError) + 重写 handle_error —— 客户端关页面刷 traceback 是噪音不是故障

六, 回归验证 (全部实跑)

脚本	结果
regress_v2.py	5/5 PASS (三大表与 store_breakeven 同源同数)
verify_layout_finance.py	11 PASS / 0 FAIL
verify_simulation_loops.py	8/8 PASS
verify_scenarios.py	21 PASS / 0 FAIL
audit_param_consumption.py	59/74 已消费, 真缺口 0

七, 口径声明与数据来源

所有经营数字均为模型推导值, 不是真实经营数据. 门店级 480 单 / ¥95,040 / EBIT −¥10,079 / 保本 1150 单 全部来自 store_breakeven 内核 + params 表, 无任何真实流水支撑. 场景里的爬坡系数, 旺季窗口, 竞品进入月份均为显式声明的假设.

参数唯一真源是 SQLite jdb.db 的 params 表 (params.json 是其体系视图). 审计期间 60 条自动应用的提案全部是 **0.0%** 同值 **no-op** (交叉验证成功), 7 条有实质变化的提案 (l01 客单价 198→268,l08 复购率,l14 CAC 等) 全部停在 **pending** 等人工确认, 参数表关键值零改动.

阻塞项: 门店连续 3 个月真实流水 (单量 / 收款额 / 新增客数 + 补充字段) 未到位. 基线采集表已交付, 拿到即出「实测 vs 推导」差异报告并回填财务基线. 真实流水校准财务基线, 访谈校准决策弹性, 调研校准竞品覆盖 —— 三者不能互相替代.