测评报告

DrawOffer 求职记忆引擎

与通用记忆方案、公开知识追踪模型的对比，102 份剧本的分层结果，以及稳定性、可追溯性与性能

测评日期	2026-10-30 至 2026-11-01
引擎版本	DrawOffer `cd08cfe4e`
测评代码与原始结果	github.com/ql-link/DrawOffer-Eval（`108cdf9`）
范围	第 1 项 正面对比、第 2 项 公开数据集、第 3 项 稳定性与可追溯性、第 4 项 剧本扩充、性能

摘要

DrawOffer 引擎读取一位求职者的简历、目标岗位、模拟面试与真实面试记录，判断他在每个知识点上的水平，指出简历说法是否夸大、经历是否讲得清楚，并预测下一道题能否答好。本报告从三方面检验它：

1. **和通用做法比**：把记录直接交给大模型，或使用开源记忆框架 Mem0。
2. **和学界模型比**：在三个公开的学生作答数据集上，与 BKT、IRT、PFA、DKT 比预测下一题。
3. **自身的可靠性**：结论能否复现、能否追溯到原话、删除资料后是否删得干净，以及推断速度。
4. **更大的剧本集**：扩到 102 份虚构求职者、581 条检查点，按方向、级别和问题类型分层，测试集封存后只打开一次。

97.7%	0.843	96.3%	0
能力判断准确率 大模型 81.4%，Mem0 81.4%	预测下一题 AUC 大模型 0.781，Mem0 0.736	结论能追溯到原话 Mem0 1.6%	删除来源后残留的结论 共删除 85 次

> 一句话：判断求职者水平、预测下一题，引擎明显好于直接用大模型；结论可复现、可追溯、删得干净，推断环节不调用大模型。差距在“经历讲得清不清楚”的判断，以及不从其他用户身上学题目难度。

1　与通用记忆方案正面对比

1.1 怎么比

三种做法读同一批记录、回答同一批检查点，用 DrawOffer 金标集同一个判分函数判分。

做法	怎么工作	模型
DrawOffer 引擎	观察器把每条记录读成结构化观察并逐字摘录原话；推断不调用模型，事先不知道会被问什么	观察器 `deepseek-v4.1-flash`
大模型读全部记录	到检查日为止的全部记录按时间排好，连同问题一起交给模型，要求结构化作答并摘录原话	`gpt-6.1-terra`（推理强度 medium）
Mem0 + 大模型	开源 `mem0ai` 2.2.1 逐条写入记录、到期删除来源；作答时每题检索 20 条记忆，只看检索结果	同上；本机嵌入 `bge-small-zh-v1.5`

数据：25 份虚构求职者剧本，共 129 个检查点。每份剧本写明各知识点随时间的真实水平、经历的真实分量和简历说法的真伪。模拟记录按产品真实流程生成：面试官与评分用产品自己的提示词，求职者由另一家厂商的模型按剧本当天的真实水平扮演。

划分：开发集 16 份，测试集 9 份。测试集在开发期间封存，2026-10-30 用 `--final` 打开一次；开封后没有再调整引擎、提示词或评测代码。

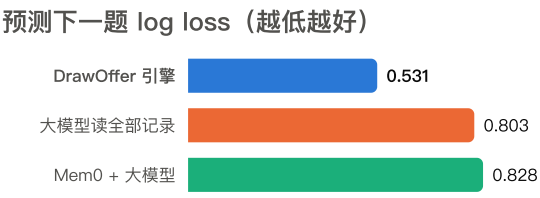
指标：

- **能力判断准确率**：`ability_level` 检查点，允许差一级；证据不足时必须答“证据不足”。
- **预测下一题**：每道知识题吸收前，预测“能通过第 L 层”的概率；标签是产品评分器的判定。AUC 区间按题重抽样 2000 次。
- **原话可追溯**：结论附带的原话，按省略号分段后全部能在原始记录里找到才算。
- **一致性**：同一输入独立重跑 5 次，每次都重新采样模型。

1.2 结果（全部 25 份剧本）

指标	DrawOffer 引擎	大模型读全部记录	Mem0 + 大模型
能力判断准确率（允许差一级）	97.7%（42/43）	81.4%（35/43）	81.4%（35/43）
预测下一题 AUC（越高越好）	0.843（0.79–0.89）	0.781（0.72–0.84）	0.736（0.67–0.80）
预测下一题 log loss（越低越好）	0.531	0.803	0.828
结论能追溯到原始记录里的原话	96.3%（520/540）	93.8%（120/128）	1.6%（2/128）
全部检查点通过率	81.4%（105/129）	84.5%（109/129）	83.0%（107/129）
同一输入 5 次，结论都相同	86.8%（112/129）	86.8%（112/129）	83.7%（108/129）
同一输入 5 次，答案逐字相同	76.0%（98/129）	70.5%（91/129）	67.4%（87/129）





统计显著性：按题重抽样（94 道题），引擎减 Mem0 的 AUC 差，95% 区间为 +0.035 到 +0.178，不含 0，优势显著；引擎减“大模型读全部记录”为 −0.005 到 +0.129，下限略低于 0，还不能说一定更好。

1.3 封存测试集（9 份剧本，只开封一次）

指标	DrawOffer 引擎	大模型读全部记录	Mem0 + 大模型
能力判断准确率（允许差一级）	100.0%（15/15）	93.3%（14/15）	86.7%（13/15）
预测下一题 AUC（越高越好）	0.809（0.72–0.89）	0.768（0.67–0.87）	0.743（0.65–0.84）
预测下一题 log loss（越低越好）	0.529	0.831	0.840
结论能追溯到原始记录里的原话	96.7%（177/183）	88.9%（40/45）	0.0%（0/45）
全部检查点通过率	88.9%（40/45）	88.9%（40/45）	84.4%（38/45）
同一输入 5 次，结论都相同	88.9%（40/45）	93.3%（42/45）	86.7%（39/45）
同一输入 5 次，答案逐字相同	80.0%（36/45）	75.6%（34/45）	71.1%（32/45）

测试集上的方向和全部数据一致：能力判断 100.0%（15/15），高于另外两种做法。测试集只有 45 个检查点，AUC 差的区间都跨过 0（引擎减大模型 −0.070 到 +0.146，引擎减 Mem0 −0.043 到 +0.174）。

1.4 按检查点类型

检查点类型	DrawOffer 引擎	大模型读全部记录	Mem0 + 大模型
能力等级	42/43	35/43	35/43
经历评估	8/14	13/14	14/14
经历合并	—	—	—
不误报问题	28/28	27/28	28/28
问题存在	15/26	21/26	20/26
问题状态	—	—	—
准备度	5/6	5/6	5/6
可补充事实	2/4	3/4	2/4

引擎领先的是能力等级（42/43，另外两种都是 35/43）。落后最多的是经历评估（8/14，大模型 13/14）和问题存在（15/26，大模型 21/26）。所以全部检查点通过率，引擎 81.4%，略低于大模型的 84.5%。

1.5 原话可追溯

引擎对人的判断，96.3%（520/540）能追溯到原始记录里的原话；大模型为 93.8%（120/128），Mem0 只有 1.6%（2/128）。Mem0 的回答引用的是它自己抽取、改写过的记忆，而不是原文：这些引用 93.0%（119/128）能在它看到的记忆里找到，但不在原始记录里。

引擎另有派生结论（准备度、被问过的题、岗位要求）230 条，其中 153 条带来源。准备度和“被问过的题”在产品数据里还没有挂接依据，这是待补的缺口。

1.6 对比中偏向基线的地方

- **基线被直接问到考点**：基线被问“X 现在什么水平”“是否存在 Y 问题”，引擎要自己发现所有问题。
- **基线模型更强**：`gpt-6.1-terra` 强于引擎观察器的 `deepseek-v4.1-flash`。
- **基线没有误报的机会**：基线只回答被问的检查点；引擎如果误报了别的问题，会在“不误报问题”检查点上扣分。

所以基线分数偏高，引擎的优势是保守估计。

1.7 成本

两个基线跑完全部对比，共调用 `gpt-6.1-terra` 1696 次（另有 396 次命中缓存），消耗 40.6 百万 token，失败 0 次。引擎的推断环节不调用模型，只有观察器读取记录时调用较小的模型。

2　公开知识追踪数据集

2.1 怎么比

知识追踪是教育领域的标准问题：根据学生过去的作答，预测下一题能否答对。我们把引擎放到三个公开数据集上，与学界常用的四个模型比较。

数据集	学生	作答	知识点	时间戳
ASSISTments 2009	4027	282,969	123	无
EdNet-KT1	5000	703,306	188	有
Junyi Academy	5000	787,452	664	有

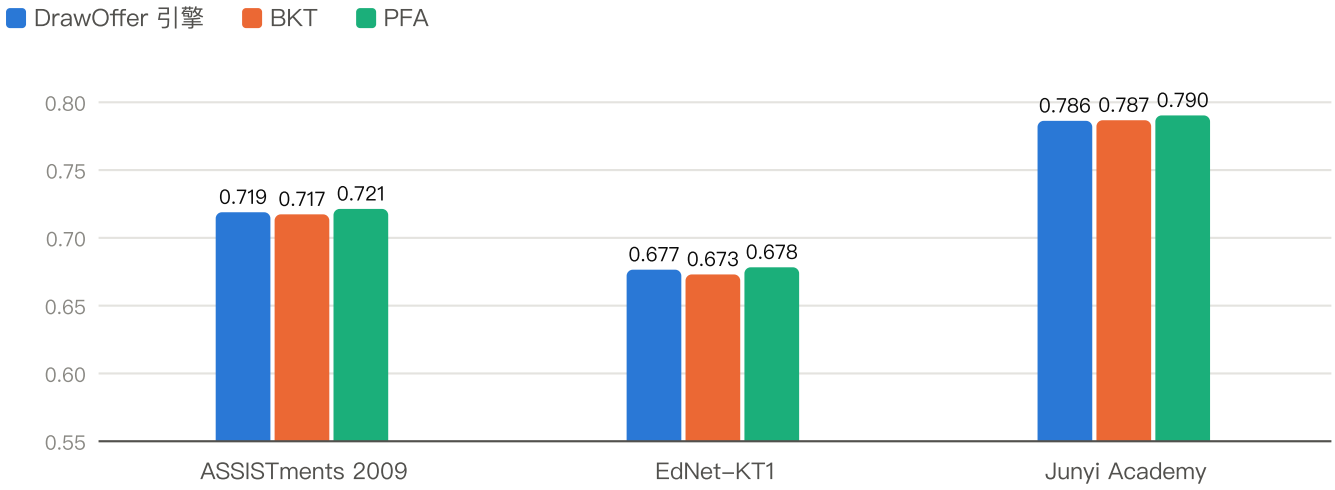
EdNet 和 Junyi 学生太多，按种子 0 从至少 20 次作答的学生中随机抽 5000 名，每人取最早的 500 次作答。三个数据集都按学生做 5 折交叉验证：每个模型只在训练折上拟合，对测试折学生的每次作答，只用他之前的作答来预测。

- **DrawOffer 引擎**（产品做法）：只从训练集估每个知识点的难度；每次作答带真实时间，预测取引擎吸收这次作答之前给出的通过概率。
- **BKT**：pyBKT 1.4.3，每个知识点一个隐马尔可夫模型。**PFA**：按成功、失败次数做逻辑回归。**IRT**：Rasch 模型，估每道题的难度。**DKT**：LSTM 深度模型。
- **对照组**：引擎关闭遗忘；引擎加上每道题的难度（产品里没有这一步）。

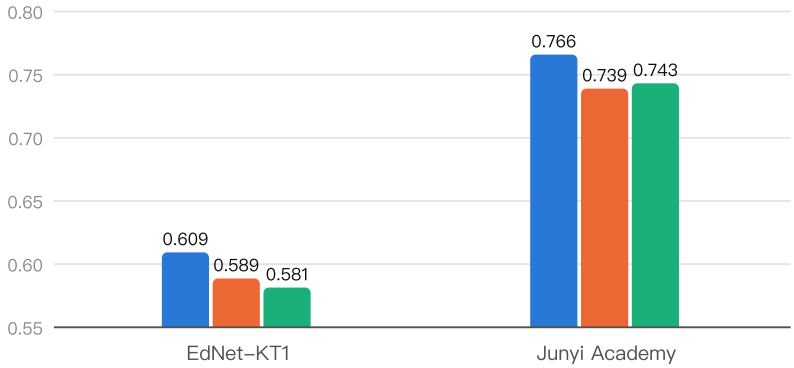
“隔天后”只统计所涉知识点距上次练习都已至少 1 天的作答，最能反映遗忘的影响。

2.2 结果

预测下一题 AUC：引擎与经典模型 BKT、PFA（纵轴从 0.55 起）



隔天后作答的 AUC（纵轴从 0.55 起；ASSISTments 没有时间戳，不参与）



- **整体与 BKT、PFA 持平**：三个数据集上，引擎与 BKT、PFA 的 AUC 差都在 ±0.005 以内。这些模型是专门为这类数据设计、并在每个数据集上单独训练的；引擎是为求职场景设计的通用推断，没有针对这些数据调参。
- **隔了一段时间再练的题，引擎更准**：EdNet 上比 BKT 高 0.021，比 PFA 高 0.028；Junyi 上比 BKT 高 0.027，比 DKT 高 0.016。
- **遗忘模型起作用**：同一个引擎打开遗忘与关闭遗忘相比，隔天后 AUC 在 EdNet 上平均高 +0.0062（5/5 折为正），Junyi 上平均高 +0.0030（5/5 折为正）。
- **和 IRT、DKT 的差距来自题目难度**：IRT 和 DKT 会从大量其他学生身上学“每道题有多难”，引擎在产品里不这样做。对照组给引擎补上题目难度后，ASSISTments 上 0.768（IRT 0.762），EdNet 上 0.758（IRT 0.763），Junyi 上 0.790（IRT 0.777）。

2.3 各数据集完整结果（5 折均值）

ASSISTments 2009

模型	AUC	RMSE	log loss
引擎 + 题目难度（对照）	0.7676 ± 0.0034	0.4244	0.5372
IRT	0.7623 ± 0.0030	0.4250	0.5381
DKT	0.7514 ± 0.0055	0.4271	0.5443
PFA	0.7213 ± 0.0067	0.4382	0.5672
DrawOffer 引擎	**0.7189 ± 0.0084**	**0.4392**	**0.5692**
BKT	0.7173 ± 0.0060	0.4390	0.5707
引擎（默认参数，不用训练集）	0.7010 ± 0.0090	0.4503	0.5919
知识点平均通过率	0.6223 ± 0.0051	0.4637	0.6197

EdNet-KT1

模型	AUC	RMSE	log loss	隔天后 AUC
IRT	0.7626 ± 0.0009	0.4345	0.5583	0.7313
引擎 + 题目难度（对照）	0.7577 ± 0.0012	0.4368	0.5627	0.7250
DKT	0.7033 ± 0.0022	0.4530	0.5979	0.6185
PFA	0.6783 ± 0.0029	0.4601	0.6124	0.5815
DrawOffer 引擎	**0.6765 ± 0.0028**	**0.4624**	**0.6169**	**0.6093**
BKT	0.6730 ± 0.0030	0.4623	0.6173	0.5886
引擎（关闭遗忘，对照）	0.6725 ± 0.0030	0.4639	0.6200	0.6031
引擎（默认参数，不用训练集）	0.6521 ± 0.0026	0.4719	0.6363	0.5921
知识点平均通过率	0.6407 ± 0.0028	0.4717	0.6362	0.5624

模型	AUC	RMSE	log loss	隔天后 AUC
DKT	0.7960 ± 0.0057	0.3317	0.3613	0.7499
引擎 + 题目难度（对照）	0.7903 ± 0.0071	0.3332	0.3653	0.7677
PFA	0.7903 ± 0.0069	0.3324	0.3632	0.7432
BKT	0.7868 ± 0.0074	0.3335	0.3708	0.7390
DrawOffer 引擎	**0.7863 ± 0.0070**	**0.3340**	**0.3664**	**0.7659**
引擎（关闭遗忘，对照）	0.7851 ± 0.0073	0.3348	0.3679	0.7629
IRT	0.7768 ± 0.0067	0.3375	0.3722	0.7648
引擎（默认参数，不用训练集）	0.7397 ± 0.0094	0.3630	0.4283	0.7028
知识点平均通过率	0.7318 ± 0.0086	0.3512	0.3981	0.7210

2.4 说明

- ASSISTments 没有时间戳，作答按顺序每题间隔 1 分钟，遗忘模型在这里不起作用。
- 遗忘与稳定度用的是产品参数，没有在这些数据上调。
- pyBKT 1.4.3 与 NumPy 2 不兼容，运行时把一个数组转成标量；它的 E 步改用 numba 编译版提速。两者预测逐位相同：有单元测试检查，ASSISTments 第 4 折复跑结果也逐位相同（1226 秒对 28 秒）。

3　稳定性与可追溯性

观察器的输出固定后（读取缓存），单独测推断本身，覆盖全部 25 份剧本。

同一输入在同一进程内推断 5 次	25/25 份剧本、60 个时点的结论逐字节相同
换 2 个新进程、不同哈希种子再推断	25/25 份剧本逐字节相同
逐一删除每份来源（85 次）	只依赖被删来源的 176 条结论全部消失；残留结论 **0** 条，仍引用被删来源的结论 **0** 条，残留观察 0 条
结论引用的原话	420/420 条能在原文中找到（按省略号分段核对；整句直接匹配 348 条）

这意味着：

- **可审计**：同一份资料永远得出同一个判断。
- **删得干净**：用户删除一份资料后，由它得出的结论随之消失，满足个人信息删除的要求。
- **有据可查**：每个判断都能指回求职者说过的原话。

注意：从读记录到出结论的端到端流程，5 次结论一致率是 86.8%（第 1.2 节）。差异来自观察器每次调用大模型都重新采样，推断这一步本身是 100% 可复现的。

4　金标集扩充到 102 份剧本

4.1 剧本与模拟

剧本覆盖 5 个方向（后端、前端、AI、产品、运营）× 4 个级别（实习、校招、中级、资深），每格 5–6 份，共 102 份、581 条检查点。 新增剧本按问题类型分配重点：简历写高或写低（技能熟练度、经历角色）、岗位知识缺口、经历先讲不清后讲清、简历与面试数字矛盾、多份简历的经历合并、复习到期与休眠、通用能力证据不足，以及表现一致、不该报任何问题的对照组。全部信息虚构。

模拟记录：新增 77 份剧本的求职者由 `claude-haiku-5.5`（经 Kiro CLI）按剧本当天的真实水平扮演；原有 25 份复用此前记录，面试官与评委用产品自己的提示词和 `deepseek-v4.1-flash`。模拟保真度（知识题评分深度与剧本一致）：开发集 50%、差一级以内 97%；测试集 56%、94%。

划分按剧本 ID 的哈希：开发集 65 份、370 条检查点；测试集 37 份、210 条，2026-11-01 用 `--final` 打开一次，开封后没有再调整引擎或剧本。

4.2 结果

80.5%
测试集检查点通过 169/210；开发集 302/370

0.64
测试集问题召回率 v2 只有 0.27

91%
测试集能力档位合理 门槛 85%

0.715
测试集预测下一题 AUC v2 0.688

指标	开发集 v3	开发集 v2	测试集 v3	测试集 v2	测试集 v3 的 95% 区间
能力档位合理	0.967	0.978	0.913	0.957	0.80–0.97
问题准确率	0.902	0.909	0.875	0.818	0.69–0.96
问题召回率	0.578	0.312	0.636	0.273	0.47–0.78
证据门槛	0.969	0.985	0.962	0.975	0.89–0.99
问题状态变化	0.273	0.000	0.125	0.000	0.02–0.47
经历去重	1.000	0.875	1.000	1.000	0.70–1.00
经历评估	0.552	0.569	0.581	0.613	0.41–0.74

v2 是上一代引擎，读同一批观察器输出。首轮测试集与开发集总通过率接近，但单次划分不足以排除过拟合。测试集上 v3 的问题召回率是 v2 的两倍多；问题准确率在测试集上高于 v2（0.875 对 0.818），开发集上持平（0.902 对 0.909）；能力档位与 v2 接近。模型标签已缓存时，单人状态推断 P95 0.054 秒，不含模型调用耗时。

4.3 按方向与级别

	开发集	测试集
后端	70/85（82%）	31/37（84%）
前端	68/73（93%）	32/40（80%）
AI	49/61（80%）	43/50（86%）
产品	55/75（73%）	29/41（71%）
运营	60/76（79%）	34/42（81%）

	开发集	测试集
实习	103/119（87%）	21/24（88%）
校招	76/97（78%）	47/49（96%）
中级	59/75（79%）	51/66（77%）
资深	64/79（81%）	50/71（70%）

产品方向最低：产品与运营的剧本主要考经历叙述，正好落在下面说的短板上。

4.4 按检查点类型

	开发集	测试集
能力档位	89/92（97%）	42/46（91%）
经历评估	32/58（55%）	18/31（58%）
经历合并	8/8（100%）	9/9（100%）
简历写高	13/20（65%）	4/5（80%）
简历写低	10/11（91%）	8/9（89%）
岗位知识缺口	7/13（54%）	4/5（80%）
数字矛盾	3/7（43%）	3/5（60%）
经历讲不清	1/9（11%）	1/6（17%）
讲不清后讲清	0/6（0%）	0/5（0%）
缺口解除或休眠	2/4（50%）	1/3（33%）
复习到期	2/3（67%）	1/3（33%）
岗位准备度	4/4（100%）	2/2（100%）
可补充事实	2/2（100%）	0/2（0%）

依赖经历评估模型的检查点，测试集通过 52/77（68%）；其余检查点 117/133（88%）。所有“不该报”的干扰项通过率都在 92% 以上，误报少。

4.5 找到并修复的问题：经历评估模型

扩充后定位到经历评估模型（读一道经历题的回答，判断本人分量、角色和每次追问讲没讲清）的两个问题：

- **分量常给“未知”，并把角色词填进分量**：修复前单票给出分量的只有 51%。把分量和角色都改成固定可选值后，单票给出分量 58%、给出角色 48%。本次只改分量的样本中，角色为 27%；最终两者一起约束。这是一次采样对比，不能单凭该结果确认变化的因果。

- **复述追问的草稿限 40 字**：超长的整条作废，放宽到 80 字。

版本（开发集重新采样对比）	通过	问题准确率	问题召回率	经历评估	单票给出分量	单票给出角色
修复前（版本 10）	292/370	0.860	0.578	0.414	51%	49%
只把分量改成可选值	300/370	0.800	0.500	0.672	75%	27%
分量和角色都改成可选值（采用）	302/370	0.902	0.578	0.552	58%	48%

表中前两版单票比例来自开发集，采用版单票比例来自开发集与测试集合计，不能直接作同样本比较；检查点通过数均来自同一开发集。M2 输出变化也会触发依赖它的数字一致性观察器新调用。

“经历讲不清”仍然几乎报不出：一是真实面试复盘入库时每题只有一组问答，追问被并进回答，判断讲清程度的模型只看追问，所以真实面试里的讲不清引擎看不到；二是部分剧本只把目标方面设成“部分讲清”，按设计不报。前者需要改复盘入库方式，属于引擎改动，尚未实施。

4.6 稳定性：同一剧本模拟 3 次

开发集 65 份剧本各模拟 3 次（复用首轮，追加两轮求职者回答及面试判定；相同请求复用缓存），370 条检查点中，322 条每次结果都相同（87%）。按多数次通过计，通过 306/370；三次各自的通过数为 302、297、307。

一致率同时包含三轮都通过与三轮都失败的检查点，不等于正确率；结果衡量剧本模拟及模型判定整条链路的波动。补跑仅补齐服务限流造成的缺失，未按分数选择结果。

5　性能

用引擎自带的生成模型造出模拟学生，每场 6 题、每两天一场，历史从 10 场到 500 场。每档 5 个学生，每人计时 3 次。

历史长度	观察条数	全量推断 P50 / P95	增量更新一场 P50	增量与全量一致
60 次作答（10 场）	120	0.005 / 0.006 秒	0.010 秒	5/5
300 次作答（50 场）	600	0.022 / 0.029 秒	0.010 秒	5/5
600 次作答（100 场）	1200	0.044 / 0.053 秒	0.012 秒	5/5
1200 次作答（200 场）	2400	0.093 / 0.207 秒	0.017 秒	5/5
3000 次作答（500 场）	6000	0.277 / 0.798 秒	0.038 秒	5/5

3000 次作答的长历史，全量推断 0.28 秒；新增一场面试，从检查点续算只要 0.038 秒，结果与全量重算逐位一致。测试机器是 Apple M3 笔记本（8 核），Darwin arm64。

6 限制

- **剧本是虚构的，由开发方起草**：第 1 节的 25 份与第 4 节的 102 份剧本都还没有经过两位独立标注人核对，模拟面试的作答由大模型按剧本扮演，偶尔会编出剧本里没有的数字。这不是真实用户数据；真实数据校准尚未开始。
- **第 1 节样本量小**：129 个检查点，测试集 45 个，所以每个数都附了区间；第 4 节扩到 581 个，但正面对比还没有在扩充后的剧本上重跑。
- **预测标签来自产品评分器**，不是人工判定；评分器本身的误差另有测评。
- **AUC 只取第 1 次运行**，一致性只在检查点上测。
- **Mem0 的实体链接没有启用**：它依赖 spaCy 英文模型，对中文不起作用。Mem0 2.2.1 只新增记忆、不合并旧记忆，这是该版本的设计。
- **公开数据集上，引擎不是第一**：整体 AUC 低于 IRT 和 DKT，原因是引擎在产品里不从其他用户身上学题目难度。

7 后续

- **在 102 份剧本上重跑正面对比**：第 1 节的大模型与 Mem0 基线目前只跑了 25 份。
- **真实数据校准**：产品已实现面试前预测存档、面试后勾选与校准报表（默认关闭），打开后即可积累数据。
- **补短板**：真实面试复盘的讲清程度判断、数字矛盾识别；给准备度和“被问过的题”挂上依据。

全部方法、代码与原始结果 JSON 见 github.com/ql-link/DrawOffer-Eval；重跑命令见该仓库 README。