Bayesian vs. Frequentist for Skill Evolution

为技能进化注入“贝叶斯后验信念”

吴晓均

xwu647@connect.hkust-gz.edu.cn
wuxiaojun@idea.edu.cn
香港科技大学（广州）

2026 年 7 月 3 日

idea

1.1 贝叶斯公式：从先验到后验

公式本身

$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)}$$

- H：我们关心的假设，例如“这个 skill 在当前环境下可靠”。
- E：新观测到的证据，例如一次验证轨迹、失败模式或输出契约结果。
- $P(H)$：先验信念；$P(E \mid H)$：如果假设成立，看到该证据的可能性。
- $P(H \mid E)$：结合证据后的后验信念。

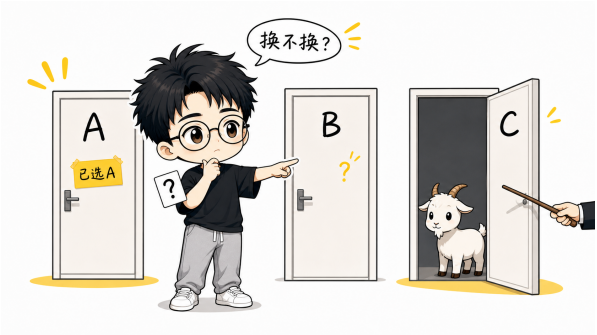
一句话直觉

贝叶斯更新不是简单“数成功次数”，而是问：在原有信念下，这个新证据应该把我们推向哪里？

1.2 三门问题：先猜一下

互动问题

- 三扇门：一辆车、两只羊。
- 你先选 A。
- 主持人知道答案，并打开 C，露出一只羊。
- 现在剩下 A 和 B：你要**坚持 A**，还是**换到 B**?



先不要看公式

请先给一个直觉判断：换门、坚持，还是无差别?

1.3 三门问题：观测会改变信念

假设 H 与证据 E

- H_A：车在 A，坚持会赢。
- H_B：车在 B，换门会赢。
- H_C：车在 C。
- E：主持人打开 C，且 C 是羊。
- 先验：
  $P(H_A) = P(H_B) = P(H_C) = 1/3$。

代入贝叶斯公式

主持人知道答案，且必须开羊门：

$$P(E \mid H_A) = \frac{1}{2}, \quad P(E \mid H_B) = 1, \quad P(E \mid H_C) = 0$$

$$P(E) = \frac{1}{2} \cdot \frac{1}{3} + 1 \cdot \frac{1}{3} + 0 \cdot \frac{1}{3} = \frac{1}{2}$$

$$P(H_B \mid E) = \frac{1 \cdot \frac{1}{3}}{\frac{1}{2}} = \frac{2}{3}, \qquad P(H_A \mid E) = \frac{\frac{1}{2} \cdot \frac{1}{3}}{\frac{1}{2}} = \frac{1}{3}$$

- 换门胜率 $2/3$；关键是 $P(E \mid H)$，不是剩两扇门。

1.4 检测阳性：先猜一下

互动问题

假设一种疾病：

- 患病率：1%。
- 灵敏度：99%。
- 假阳性率：5%。
- 如果某人检测阳性，他真正患病的概率是多少?

请先估计

这个后验概率会接近 99% 吗？为什么？

1.5 检测阳性：基础率不能被忽略

假设 H 与证据 E

- H：这个人真的患病。
- E：检测结果为阳性。
- 先验：$P(H) = 0.01$。
- 似然：$P(E \mid H) = 0.99$。
- 假阳性：$P(E \mid \neg H) = 0.05$。
- 要算的是 $P(H \mid E)$，不是 $P(E \mid H)$。

代入贝叶斯公式

$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)}$$

$$P(E) = P(E \mid H)P(H) + P(E \mid \neg H)P(\neg H)$$

$$P(E) = 0.99 \times 0.01 + 0.05 \times 0.99 = 0.0594$$

$$P(H \mid E) = \frac{0.99 \times 0.01}{0.0594} \approx 16.7\%$$

- 阳性不是 99% 患病，因为先验基础率很低。
- 对技能也是一样：一次成功不等于技能可靠。

2.1 今天的核心问题

LLM Agent 的能力越来越来自“外部推理环境”

- 不只是模型参数 θ，还包括 prompts、tools、memory、SOPs、skills、harness feedback。
- 同一个模型在不同 C 下运行，实际采样更像 $P(X \mid \theta, C)$。
- 因此，能力维护的对象从“调模型”扩展为“维护模型运行时的条件”。

本文观点

- reusable skills / SOPs 不是静态文本，而是关于未来任务成功率的不确定假设。
- 技能进化不应只是“成功次数 / 总次数”，而应维护可累计、可审计、可迁移的后验信念。

2.2 朴素频率式修补为什么危险

频率学派式维护

- 观察失败
- 统计成功率
- 追加自然语言 patch
- 下一轮从近似“无状态”计数继续

Agent 轨迹的现实

- 样本少：每个任务轨迹都贵
- 依赖强：上下文、工具、验证器共同决定结果
- 失败异质：一次失败可能是偶然，也可能是可复用故障模式
- 早期极端值会诱导错误 patch、过早 split 或 retire

2.3 映射回技能进化

Bayesian-Agent 的一句话

把每个 reusable skill / SOP 看作一个带不确定性的假设：它在某个模型、上下文、工具接口和验证器下，到底有多可靠？

频率式视角

- 成功率是点估计：$\hat{p} = S/n$。
- 少样本时容易跳到 0 或 1。
- 难以解释为什么 patch、split、compress、retire。

贝叶斯视角

- 先验 + 验证轨迹 + 失败模式 + 成本 + 上下文。
- 得到 posterior belief，而不是裸成功率。
- 每次技能动作都能回溯到证据链。

3.1 研究对象：外部技能，而不是模型参数

固定模型，优化推理环境

$$p_{k,t} = P(y_t = 1 \mid M_\theta, C_t, h_k, z_t)$$

- M_θ：冻结的 LLM / agentic model。
- C_t：prompt、retrieved context、tool/action interface、verifier feedback。
- h_k：可复用 skill / SOP / failure-mode patch。
- $z_t = g(e_t)$：从验证轨迹中抽取的离散特征签名。

核心转向

不是问“模型会不会做这道题”，而是问“这个外部 skill 在当前 harness 条件下是否值得继续影响未来运行”。

3.2 频率估计可以诊断，但不适合作为决策状态

Empirical-frequency

$$q_F = \frac{S}{n}$$

- 一次成功：$q_F = 1$。
- 一次失败：$q_F = 0$。
- 没有观测时还需要额外 backoff。

Beta-Bernoulli posterior

$$q_B = \frac{\alpha_0 + S}{\alpha_0 + \beta_0 + n}$$

- 默认 $\alpha_0 = \beta_0 = 1$。
- 一次成功：$q_B = 2/3$，不是 1。
- 一次失败：$q_B = 1/3$，不是 0。
- 早期证据被平滑，后续证据逐渐主导。

3.3 后验风险：为什么后验均值是更合适的信念

局部 skill cell：$\omega = (h_k, z)$

假设 θ_ω 是该 skill cell 的真实成功概率，使用平方预测损失：

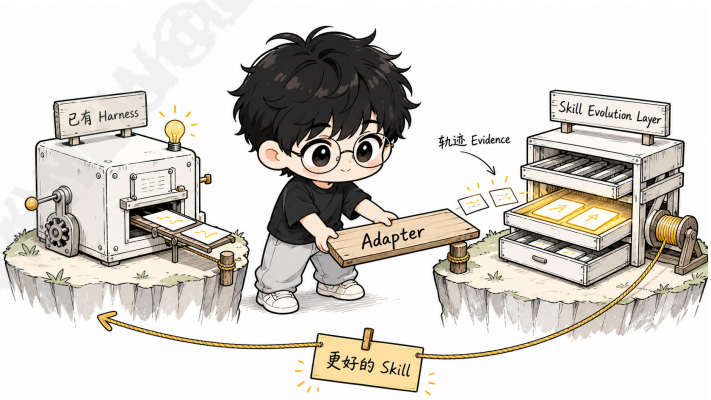
$$L(q, \theta_\omega) = (q - \theta_\omega)^2.$$

论文中的有限样本结论

$$\mathbb{E}[L(q_F, \theta_\omega) \mid S, n] - \mathbb{E}[L(q_B, \theta_\omega) \mid S, n] = (q_F - q_B)^2 \geq 0.$$

- 在 harness 自己的 uncertainty model 下，posterior mean 最小化后验期望风险。
- 这不是说频率估计“永远错”，而是少样本技能决策不应被裸计数牵着走。

4.1 整体框架



- **Full mode**：registry 从空开始，完整 benchmark 在线演化。
- **Incremental mode**：先读已有 agent run，再通过 adapter 只重跑失败任务，测量 plug-in repair。

4.2 证据从哪里来

只用验证过的轨迹更新 belief

$$e_t = (x_t, h_k, c_t, y_t, u_t, \tau_t, \ell_t, r_t, m_t)$$

- y_t：来自 benchmark grader 或 output contract，而不是模型自评。
- u_t, τ_t, ℓ_t：token、turn、latency 等成本信号。
- r_t：验证器抽取的 failure mode，例如 missing output、format error。

离散化特征签名

$$z_t = g(e_t) = (c_t, r_t, b_u(u_t), b_\tau(\tau_t), b_\ell(\ell_t), m_t^{\leq 80})$$

b_u, b_τ, b_ℓ：分桶函数；$m_t^{\leq 80}$：只保留长度 ≤ 80 的短 metadata。

4.3 后验证据模型

类别先验

$$\pi_{k,t}(\ell) = \frac{N_{k,\ell} + \lambda}{\sum_{\ell'} N_{k,\ell'} + \lambda|\mathcal{Y}|}$$

成功后验

$$s_{k,t}(z) = \frac{\tilde{p}_{k,t}(1 \mid z)}{\tilde{p}_{k,t}(0 \mid z) + \tilde{p}_{k,t}(1 \mid z)}$$

特征似然

$$\theta^{(\ell)}_{k,j,t}(v) = \frac{N_{k,j,\ell,v} + \lambda}{\sum_{v'} N_{k,j,\ell,v'} + \lambda|\mathcal{V} \cup \{v\}|}$$

- 默认使用 factorized categorical likelihood。
- 所有乘积在 log space 中计算。
- Beta-Bernoulli summary 保留用于 audit 与保守失败判断。

记号：$\ell \in \{0, 1\}$ 是失败/成功标签，N 是计数，$\lambda = 1$ 是 Laplace smoothing，$\tilde{p}$ 是未归一化分数。

4.4 五类 posterior-guided skill actions

动作	默认触发条件与含义
Explore	没有观测，或后验仍不确定；继续收集证据。
Patch	同一 failure mode 至少出现 2 次；把失败模式转成 guardrail。
Split	至少 3 个上下文、4 条观测；一个宽泛 skill 可能覆盖异质场景。
Compress	至少 3 条观测且成功后验 ≥ 0.72；可靠 skill 保持简洁。
Retire	失败证据占优：$\beta \geq 4$ 且成功后验 < 0.45。

注：阈值是默认工程 policy；β 是 Beta-Bernoulli 中失败侧后验参数。

改写自论文 Table 2。

4.5 重要设计：后验审计与模型提示分离

Posterior audit

- 成功后验与上下文条件成功率
- 成本、失败模式、观测次数
- before/after skill snapshot
- 用于 ranking、debugging、解释决策

Model-facing skill text

- 不把一串 posterior 数字塞给 LLM。
- 只渲染可执行 guardrails 与 failure-mode patches。
- 例如：终止前检查输出文件、过滤空 OHLCV 行、验证格式。

结果

Agent 看到的是可执行 SOP；研究者看到的是可审计证据链。

5.1 实验设置

Benchmark

- SOP-Bench：多步骤工业 SOP 执行。
- Lifelong AgentBench：顺序任务与跨任务经验复用。
- RealFin-Bench：金融推理、隐含前提、文件输出与验证约束。

比较对象

- GA：GenericAgent baseline。
- BA-Full：空 registry 在线演化。
- BA-Inc：吸收 GA 轨迹，只重跑失败任务。
- 额外比较：frequentist control、backend ladder、model scaling。

指标

task accuracy、input/output/total tokens、每百万 token 解决任务数。

5.2 Bayesian vs. Frequentist：RealFin 控制实验

Backend	Model	Method	Accuracy	Success	Input	Output	Total	Efficiency
GA	flash	Bayesian	52.5%	21/40	2.90M	244k	3.15M	6.67
GA	flash	Frequentist	52.5%	21/40	3.06M	244k	3.31M	6.35
GA	pro	Bayesian	65.0%	26/40	3.38M	323k	3.70M	7.02
GA	pro	Frequentist	57.5%	23/40	3.46M	309k	3.77M	6.10
Native BA	flash	Bayesian	70.0%	28/40	10.43M	463k	10.89M	2.57
Native BA	flash	Frequentist	37.5%	15/40	3.79M	236k	4.03M	3.72
Native BA	pro	Bayesian	70.0%	28/40	9.33M	579k	9.91M	2.83
Native BA	pro	Frequentist	45.0%	18/40	3.86M	299k	4.16M	4.32

复制自论文 Table 3：Bayesian vs. frequentist skill evolution on RealFin-Bench.

5.3 控制实验图：准确率与 token cost

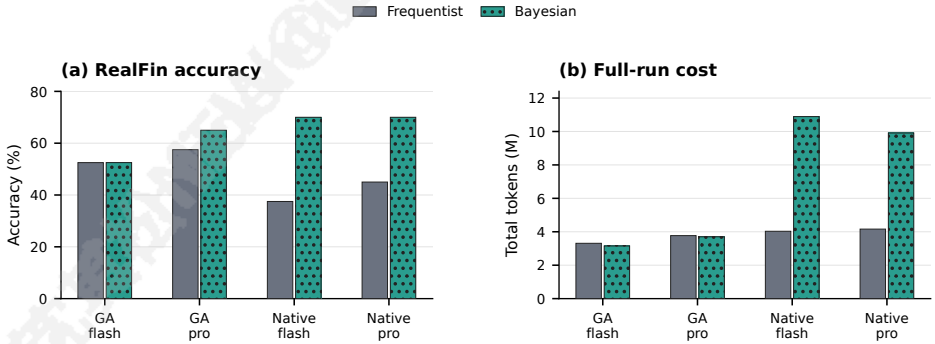


图 1: Bayesian-vs.-frequentist RealFin control.

- Native backend 上差距最明显：flash 为 28/40 vs. 15/40，pro 为 28/40 vs. 18/40。
- Bayesian rows 更贵，但更多 token 变成了可验证输出，而不是早停失败。

5.4 RealFin case study：差距来自哪里

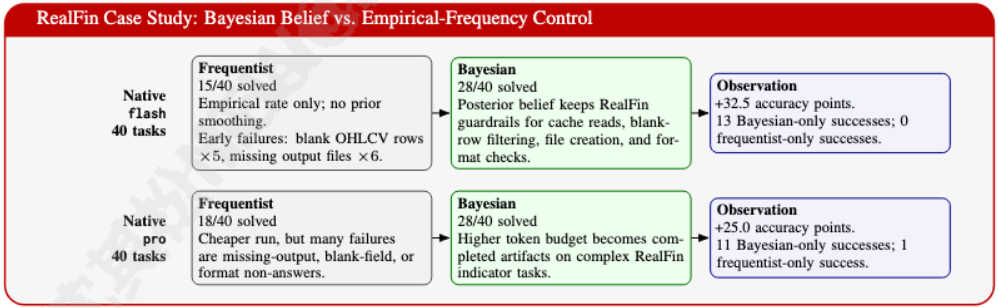


Figure 4: Task-level case study for the native RealFin Bayesian-vs.-frequentist comparison. The frequentist controller is cheaper but often stops with missing output files or blank-field crashes. Bayesian-Agent spends more inference budget while retaining posterior-weighted execution guardrails, yielding substantially higher completion on these sparse, expensive tasks.

5.5 理论证明：后验均值降低后验风险

1. 局部 skill cell. 对 $\omega = (h_k, z)$，令 θ_ω 表示真实成功概率：

$$y_i \mid \theta_\omega \sim \text{Bernoulli}(\theta_\omega), \quad \theta_\omega \sim \text{Beta}(\alpha_0, \beta_0), \quad S = \sum_{i=1}^{n} y_i.$$

2. 频率点估计

$$q_F = \frac{S}{n}$$

少样本时可被一次成功/失败推到 1 或 0。

3. 贝叶斯后验均值

$$q_B = \mathbb{E}[\theta_\omega \mid D] = \frac{\alpha_0 + S}{\alpha_0 + \beta_0 + n}$$

先验会平滑早期偶然证据。

4. 平方损失下的后验风险分解

$$\rho(q \mid D) = \mathbb{E}[(q - \theta_\omega)^2 \mid D] = \text{Var}(\theta_\omega \mid D) + (q - q_B)^2.$$

$$\boxed{\rho(q_F \mid D) - \rho(q_B \mid D) = (q_F - q_B)^2 \geq 0}$$

- 理论上：在该后验风险模型下，Bayesian posterior mean 不劣于经验频率点估计。
- 实验上：RealFin case study 显示这种优势转化为更稳的 guardrails 与更高 completion。

5.6 主结果：DeepSeek 上的 GA / BA-Full / BA-Inc

Benchmark	Agent	Model	Accuracy	Total Tokens	Efficiency	主要变化
SOP	GA	flash	80%	1.39M	11.47	baseline
SOP	BA-Full	flash	95%	1.26M	15.13	+15 points
SOP	BA-Inc	flash	95%	153k	19.63	repair-only
Lifelong	GA	flash	90%	690k	26.07	baseline
Lifelong	BA-Full	flash	85%	674k	25.23	online negative case
Lifelong	BA-Inc	flash	100%	84k	23.85	fixes 2/2 failures
RealFin	GA	flash	45%	4.24M	4.24	baseline
RealFin	BA-Full	flash	52%	3.15M	6.67	+7 points
RealFin	BA-Inc	flash	65%	2.02M	3.96	converts 8/22 failures
RealFin	GA	pro	60%	3.72M	6.46	baseline
RealFin	BA-Full	pro	65%	3.70M	7.02	+5 points
RealFin	BA-Inc	pro	68%	1.72M	1.74	converts 3/16 failures

从论文 Table 4 摘取 DeepSeek 与 Bayesian-Agent 相关行。

5.7 主结果图：增量修复最稳定

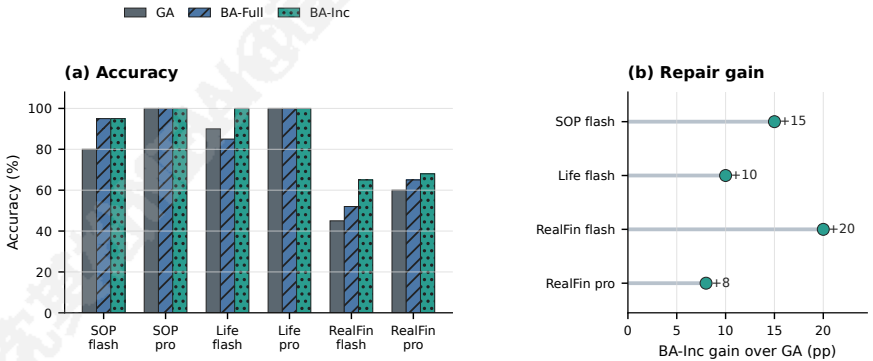


图 2: Visual analysis of Bayesian-Agent on DeepSeek backbones.

- BA-Inc 在非零修复设置中都提升最终 accuracy。
- BA-Full 有负例：Lifelong flash 从 90% 降到 85%，说明在线演化会受稀疏证据与任务顺序影响。

5.8 Backend ablation：更复杂 harness 上是否仍有效

Backend	Model	SOP-Bench			Lifelong			RealFin		
		Base	Full	Inc	Base	Full	Inc	Base	Full	Inc
BA native	flash	95%	100%	100%	95%	100%	100%	62.5%	70%	72.5%
BA native	pro	100%	100%	100%	100%	100%	100%	65%	70%	77.5%
MiniSWEAgent	flash	100%	95%	100%	85%	95%	100%	60%	55%	70%
MiniSWEAgent	pro	95%	100%	100%	90%	100%	100%	70%	70%	80%
Claude Code	flash	90%	100%	100%	100%	100%	100%	77.5%	80%	87.5%
Claude Code	pro	65%	95%	100%	100%	100%	100%	65%	67.5%	75%

复制自论文 Table 5。

5.9 Backend 图：Bayesian layer 可迁移，但 cost 不同

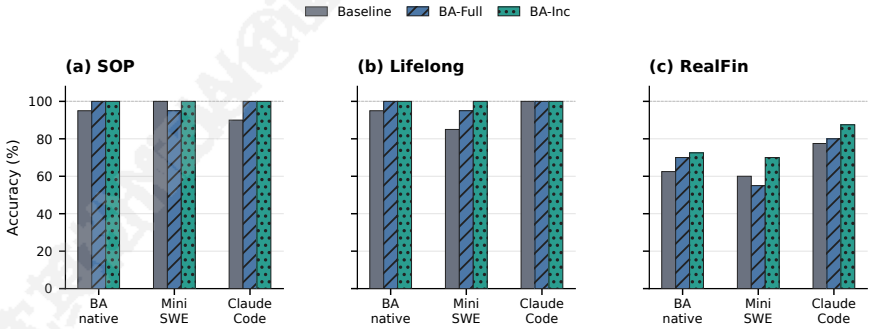


图 3: Backend ablation on `deepseek-v4-flash`.

- RealFin 上 BA-Inc 从不低于对应 baseline。
- Claude Code + flash 达到最高 RealFin：87.5%，但 token 成本最高。
- MiniSWEAgent + pro 达到 80%，是更好的 accuracy–cost tradeoff。

5.10 Model scaling：模型变强只有在 contract 成立时才有用

Model	Provider	Size	Success	Accuracy	Input	Output	Total	Efficiency	No File / Invalid
qwen3.5-35b-a3b	DashScope	35B/A3B	18/40	45.0%	6.95M	370k	7.32M	2.46	10 / 12
qwen3.5-122b-a10b	DashScope	122B/A10B	3/40	7.5%	3.19M	144k	3.34M	0.90	37 / 0
deepseek-v4-flash	DeepSeek	284B	22/40	55.0%	6.50M	453k	6.96M	3.16	9 / 9
deepseek-v4-pro	DeepSeek	1.6T	28/40	70.0%	6.28M	459k	6.73M	4.16	7 / 5

注：A3B/A10B 为模型标识中的 active-parameter shorthand；No File= 未生成输出文件，Invalid= 生成文件但未通过 verifier。

复制自论文 Table 6。

5.11 Scaling 图：异常点本质是输出契约失败

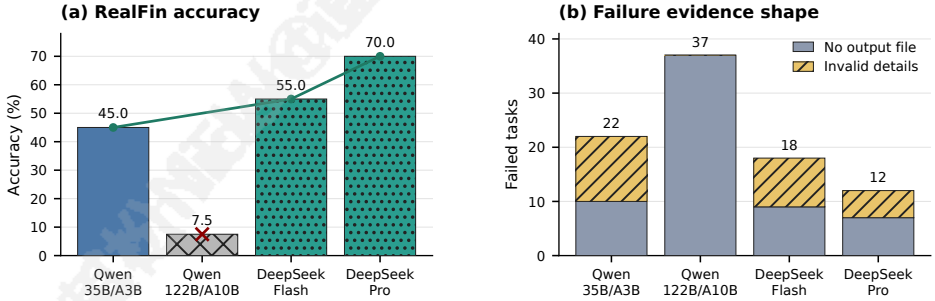


图 4: Model scaling on RealFin-Bench under MiniSWEAgent.

- 主路径：Qwen 35B-A3B → DeepSeek Flash → DeepSeek Pro，准确率 45.0% → 55.0% → 70.0%。
- Qwen 122B-A10B 的 37/40 failures 都是 no-file：这更像 adapter/file-output contract 失败，而不是纯推理能力不足。

6.1 Skill evolution trace：三类 action 的代表轨迹

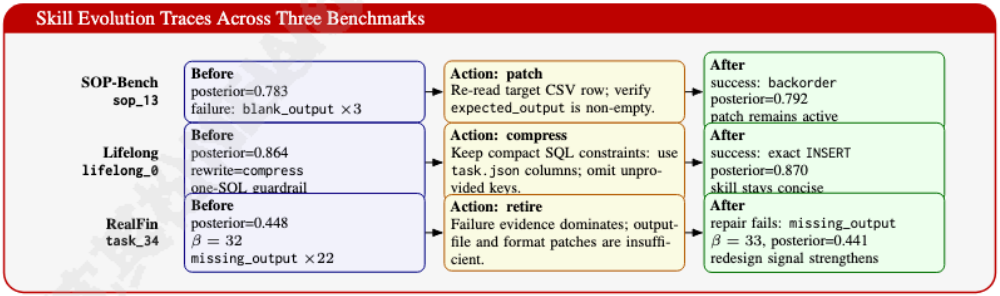


Figure 5: Representative skill-evolution traces. SOP-Bench shows a recurring failure mode becoming a concrete patch, Lifelong AgentBench shows stable evidence leading to compact skill context, and RealFin-Bench shows a negative case where repeated output-file failures strengthen a retire/redesign decision rather than being hidden.

6.2 技能文本案例：before / after trace

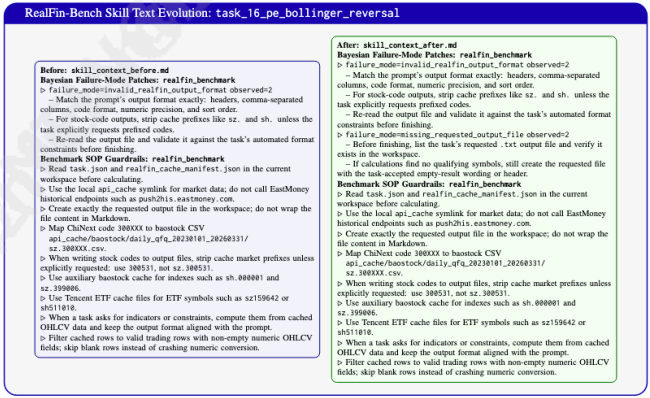


Figure 8: Before/after model-facing skill text for RealFin-Bench. The after-state adds a missing-output-file patch, making file creation and empty-result handling explicit in addition to the existing output-format patch and data-cache guardrails.

6.3 什么时候 Bayesian-Agent 最有用

适合

- 任务会重复出现，且每次运行代价高。
- 有验证器或 output contract。
- 失败模式会复现，例如格式、文件、工具调用、缓存路径。
- harness 能接收 skill / SOP 文本作为干预位置。

不适合或收益有限

- 一次性任务，没有可迁移证据。
- 主观标签或验证器缺失。
- 环境强非平稳，历史证据迅速过期。
- 失败来自缺工具、缺数据，轨迹无法形成可修复信号。

6.4 Takeaways

1. 技能不是“文本补丁”，而是 evidence-bearing hypotheses

skills / SOPs 的可靠性应由先验、验证轨迹、失败模式、成本与上下文共同更新。

2. 后验信念让少样本 agent 维护更稳

在稀疏、昂贵、上下文依赖的轨迹中，posterior belief 比裸成功率更适合作为持久技能动作的决策状态。

3. Bayesian-Agent 把技能进化变成 harness optimization

patch、split、compress、retire 不再是无状态修补，而是 posterior-guided、evidence-calibrated、可审计的演化过程。

6.5 对 Agent Skills 系统开发的启示

按证据规模选择 skill evolving 机制

- 数据量较大、反馈稳定、可重复采样：可以采用 Skill-Opt、SkillGrad 这类“训练式”方法，直接优化 skill / prompt / policy。
- 少样本、轨迹昂贵、上下文强依赖：优先采用 Bayesian-Agent，把 skill 视为带不确定性的假设，并维护 posterior belief。
- 关键不是二选一，而是让证据规模与任务成本决定更新方式：多样本适合训练，少样本需要校准信念。

6.6 Selected References

This talk

- Wu et al. 2026. *Bayesian-Agent: Posterior-Guided Skill Evolution for LLM Agent Harnesses*. ACL submission.

Skill evolution

- Liang et al. 2026. *GenericAgent*. arXiv:2604.17091.
- Yang et al. 2026. *SkillOpt*. arXiv:2605.23904.
- Wang et al. 2026. *SkillGrad*. arXiv:2605.27760.

Benchmarks

- Nandi et al. 2025. *SOP-Bench*. arXiv:2506.08119.
- Zheng et al. 2025. *LifelongAgentBench*. arXiv:2505.11942.
- Dai et al. 2026. *RealFin*. arXiv:2602.07096.

Bayesian foundations

- Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. MIT Press.
- Shahriari et al. 2016. *A Review of Bayesian Optimization*. Proc. IEEE.
- Frazier. 2018. *A Tutorial on Bayesian Optimization*. arXiv:1807.02811.

完整引用见 Bayesian-Agent 论文 bibliography。

感谢聆听！

吴晓均

xwu647@connect.hkust-gz.edu.cn
wuxiaojun@idea.edu.cn

idea