

Bottling as an Investment Decision: An Analytical Reanalysis of Cost–Quality Trade-offs in the BOTTLED Benchmark

Bottling as an Investment Decision: An Analytical Reanalysis of Cost–Quality Trade-offs in the BOTTLED Benchmark

Abstract

Organisations that run millions of similar queries through large language model (LLM) APIs pay per instance, and the decision to invest in a cheaper task-specific artifact is today made by hand, if at all. The BOTTLED benchmark shows that LLM agents can make this investment autonomously — a capability its authors call "bottling" — but that strong zero-shot performance does not reliably transfer: 48 of 60 bottling runs fall below the lower bound of the 95% confidence interval of their model's zero-shot score, and 31 of 60 underperform a small-model distillation baseline. What the reported picture lacks is an explicit decision theory: when is bottling worth the setup cost, and how surprising are the observed failure rates? This paper supplies that analysis using only arithmetic on the published figures. Under an independence assumption, the probability of observing 48 or more of 60 runs below a per-model 95% confidence floor is approximately 1.3×10^{-65} , so bottling failure is a structural, model-level property rather than per-run noise; by contrast, 31 of 60 runs below a distillation baseline is statistically consistent with a coin flip ($z \approx 0.26$, two-sided $p \approx 0.80$). On the reported successes, Opus 5's bottled artifact delivers approximately 539 times more macro-F1 per unit of per-instance cost than direct querying, and 3.76 times more than the purpose-built "system one" model Jev, with a quality-adjusted break-even workload of $n_\rho^* \approx 1.22 C_0/c_L$. The analysis converts BOTTLED's empirical findings into a usable investment rule, a null model for auditing future agent results, and a governance agenda for agent-created artifacts.

1. Introduction

A team operating a production LLM pipeline — classifying millions of query-product pairs, routing support tickets, labelling documents — pays per query. At per-instance API pricing, a workload of 10^6 instances at even a modest per-query cost becomes a recurring line item that scales linearly with volume, and the standard remedies (prompt compression, smaller models, distillation) each require engineering effort that must be justified against the workload before it is spent. Today that justification is a human project decision: someone estimates volume, prototypes a small model or a reusable program, validates quality, and only then switches the workload over. The decision is slow, and for the long tail of medium-sized repetitive workloads it is often never made at all, so those workloads stay on expensive per-query inference indefinitely.

LLM agents change what is possible here. If an agent can receive an entire unlabelled workload and autonomously produce a cheaper task-specific solution — training a small model, or writing a reusable program — then the investment decision itself can be automated, and the long tail of repetitive workloads becomes addressable. The BOTTLED benchmark [1], [2] measures exactly this ability, which its authors name "bottling": turning general capabilities into task-specific solutions that balance answer quality and amortised cost, under fixed time, compute and LLM API budgets. Its headline findings cut both ways. Bottling can yield substantial savings — on query-product relevance classification, Opus 5 retains about 82% of its zero-shot macro-F1 at roughly 657 times lower reported cost, and recovers about 94% of the macro-F1 of Jev, a "system one" model built for cheap, repetitive inference, at a quarter of Jev's projected full-workload cost. But strong zero-shot performance does not reliably

translate into strong bottling: models with similar zero-shot scores can differ substantially after bottling, 48 of 60 bottling runs score below the lower bound of the 95% confidence interval of their model's zero-shot performance, and 31 of 60 runs underperform the stronger of two small-model distillation baselines with the same token budget.

What the benchmark report does not supply, and what a deployment decision needs, is the decision theory underneath the numbers. A practitioner reading "48 of 60 runs below the zero-shot confidence floor" cannot tell whether that is mild per-run noise or a structural property of current models; a practitioner reading "657 times cheaper" cannot tell at what workload size the investment pays off once quality loss is priced in. This paper provides both, using only arithmetic on the reported figures and a stated cost model. If the analysis holds, an operator can read any bottling result as an investment appraisal: a break-even workload, a quality-adjusted cost ratio, and a falsifiable expectation for how failure rates should behave if bottling skill were randomly distributed across runs.

The contributions are:

- A formal cost-and-quality model of bottling as an investment problem, with raw and quality-adjusted break-even workload formulas, derived in Sections 3 and 4.6.
- An exact binomial reanalysis showing that the observed 48-of-60 failure count is inconsistent with independent per-run failures at any plausible rate (tail probability approximately 1.3×10^{-65} under the 95% confidence-floor reading; $z \approx 38.4$ by normal approximation), derived in Section 4.2.
- A contrasting analysis showing that 31 of 60 runs below the distillation baseline carries almost no evidence against a symmetric null ($z \approx 0.26$, two-sided $p \approx 0.80$), derived in Section 4.3.
- Computed quality-per-cost figures for the reported successes — $B = 0.82 \times 657 = 538.74$ against direct querying and $B = 3.76$ against Jev — with the minimum retention threshold $r_{\min} = 1/657 \approx 0.00152$ and sensitivity analysis, derived in Sections 4.5 and 4.6.
- A practical adoption pathway, readiness assessment, and governance agenda for agent-created artifacts, given in Section 7.

2. Background and Related Work

The direct subject of this paper is the BOTTLED benchmark and its underlying paper, "Agent in a Bottle: Can LLM Agents Turn Their Capabilities Into Cheap, Scalable Artifacts?" [1], [2]. The supplied abstract states the problem and the experimental design: LLM agents receive an entire unlabelled workload and must complete it under fixed time, compute and LLM API budgets, choosing their own approach such as training a small model or writing a reusable program. Across ten models and three tasks, strong zero-shot task performance does not reliably translate into strong bottling capability; 48 of 60 bottling runs score below the lower bound of the 95% confidence interval of their model's zero-shot performance, and 31 of 60 runs underperform the stronger of two small-model distillation baselines with the same token budget. The same abstract reports the positive results this paper quantifies: Opus 5 retains about 82% of its zero-shot macro-F1 on query-product relevance classification at roughly 657 times lower reported cost, and recovers about 94% of the macro-F1 of Jev, a "system one" model built for cheap, repetitive inference, at a quarter of Jev's projected full-workload cost. Entries [1] and [2] are two records of the same work — a query record and a titled record — and this paper treats them as one source, citing both where the benchmark's design and numbers are invoked. The supplied abstract gives no further detail on the tasks' construction, the confidence-interval estimation procedure, or the distillation baselines' architecture; this paper therefore treats those quantities as reported summary statistics and derives everything else arithmetically.

The closest conceptual precursor in the supplied bibliography is the plan-reuse mechanism for LLM-driven agents [9]. Its abstract describes LLM-driven personal assistants that receive user requests, generate plans using an LLM, execute those plans through tools, and return results; the contribution is a mechanism for reusing plans across

requests. Plan reuse and bottling share the same economic instinct — pay once for a structure, then replay it cheaply — but plan reuse operates at the granularity of a single request's plan, whereas bottling operates at the granularity of an entire workload's solution. The supplied summary of [9] gives no further detail on the mechanism's internals, so this paper draws only the granularity contrast and cites it as prior art in kind, not in magnitude.

Work on conventions and shared reusable structures in multi-agent reinforcement learning supplies a second parallel. The Hanabi study [7] states that the card game is a strong medium for testing and developing multi-agent reinforcement learning algorithms because of its cooperative nature, partial observability, limited communication and complexity, and that augmenting the action space with conventions improves multi-agent cooperation. A convention is a compressed, reusable coordination artifact; bottling asks whether an LLM agent can manufacture the analogous artifact for a workload. The entry reports no quantitative outcome we may cite. The survey of multi-agent deep reinforcement learning with communication [3] frames communication as an effective mechanism for coordinating agent behaviours, broadening their views of the environment and supporting collaboration, with agents exchanging various types of messages. In bottling terms, the "message" an agent sends to its own future is the artifact itself — a trained model or a program — and the survey's framing of communication as performance-improving is the multi-agent analogue of the amortisation argument formalised in Section 3; the entry says nothing about LLM artifact creation, and no finding beyond that framing is drawn from it.

The semantic-web line of work [6] addresses inter-agent communication at the semantic level, introducing an efficient mechanism for discovering semantic relationships among agent communication protocols constituted of communication acts. Its relevance here is structural: if agents autonomously produce artifacts that other systems will consume, the artifacts need discoverable, machine-readable descriptions of what they do, and [6] is the supplied bibliography's only entry addressing protocol-level semantic relationships. The entry's summary, as supplied, does not state the mechanism's features or results in detail, so no performance claim is drawn from it.

A complementary application-side precedent is Focus Agent [5], an LLM-powered framework that, in the human-computer interaction domain, simulates both the focus group for data collection and acts as a moderator, motivated by the observation that focus groups are resource-intensive and demand skilled moderators and meticulous preparation. Focus Agent substitutes an LLM framework for an expensive human process; bottling substitutes a cheap artifact for an expensive per-instance inference process. Both are instances of the same substitution logic, and the Focus Agent summary's emphasis on preparation effort parallels this paper's setup-cost term C_0 ; the entry describes no cost accounting, so it is used only as a motivational parallel.

The risk literature supplies the governance motivation. "Agents at Risk" [4] states that LLM-based agents are increasingly deployed in applications such as trip-planning and web-use agents, and that prior work has shown agents are vulnerable to context confusion, where external adversarial content incorporated into the agent's reasoning context may be treated as task-relevant constraints. An agent that autonomously writes and deploys a reusable artifact amplifies this exposure: an artifact trained or programmed under context confusion encodes the confusion into every downstream instance. The supplied summary of [4] states that LLM-based agents are increasingly deployed in applications such as trip-planning and web-use agents and are vulnerable to context confusion, where external adversarial content incorporated into the agent's reasoning context may be treated as task-relevant constraints; the title further indicates the work concerns users unwittingly undermining LLM safety, which bears on who supervises the bottling run. The entry supplies no quantification of this risk, and it is flagged as an open question in Section 6. The value-systems work [8] frames Agreement Technologies as open computer systems in which autonomous software agents interact on behalf of humans to reach mutually acceptable agreements, and argues that with the advance of AI systems such agreements must remain aligned with ethical principles and moral values; it studies learning the value systems of agents with preference-based and inverse reinforcement learning. A

bottled artifact is an agreement of sorts — a commitment about how a workload will be answered — and [8] supplies the supplied bibliography's only vocabulary for asking whether that commitment tracks the principal's values.

Two QNFO entries connect the analysis to model selection and to research governance. The LiveBench-grounded audit [10] reports an August 2026 audit of the frontier LLM landscape for mathematics-heavy scientific research and agentic AI, grounded in the contamination-free LiveBench 2026-06-25 benchmark, finding that no Llama model ranks in the top 42 and naming DeepSeek V4 Pro 0813 as the mathematics price-performance outlier; the supplied summary is truncated mid-sentence at that point, so no further finding is drawn from it. Because bottling outcomes differ substantially across models with similar zero-shot scores [1], [2], selection criteria that look only at benchmark rank — the kind [10] audits — are insufficient for choosing a bottling agent, which motivates the two-axis reading developed in Section 4. The epistemic-legibility position paper [11] argues that as AI systems take on hypothesis generation, experiment planning, analysis and writing, the scientific ecosystem's bottleneck shifts from production to evaluation, verification and governance, and contributes a documented case and an ignorance-auditing instrument to that problem. Bottled artifacts are precisely the kind of agent-produced output that [11]'s governance instrument targets: an artifact's provenance, validation budget and failure mode should be auditable before a workload is switched over. The flag-structures survey [12] supplies, in its summary, a family of mathematically precise objects — flag varieties, flag matroids, flag algebras, Finsler directional curvature invariants — and advertises explicit counting derivations; its methodological parallel to this paper is the insistence on explicit counting, which Section 4 follows, though its subject matter does not otherwise bear on bottling. Finally, the entry "Agentic Collapse" [13] supplies a title with no abstract text; no substantive claim is drawn from it, and it is listed here only to document that the supplied bibliography was consulted in full.

3. Methods

3.1 The bottling investment problem

Let a workload consist of n instances. Two execution paths exist. The direct path queries a large model on every instance at per-instance cost c_L , yielding quality Q_L (normalised to $Q_L = 1$). The bottled path first invests a one-time setup cost C_0 — tokens, compute and time spent building the artifact, such as training a small model or writing a reusable program, within the benchmark's fixed budgets [1], [2] — and then serves each instance at per-instance cost $c_A < c_L$, yielding quality $Q_A = \rho Q_L$ with retention $0 \leq \rho \leq 1$.

The two paths' total costs are

$$C_{\text{direct}}(n) = n c_L, \quad C_{\text{bottled}}(n) = C_0 + n c_A.$$

The raw break-even workload n^* solves $C_{\text{bottled}}(n^*) = C_{\text{direct}}(n^*)$:

$$n^* = \frac{C_0}{c_L - c_A}.$$

Quality-adjusted break-even instead equalises cost per unit of quality. The direct path costs c_L per unit quality; the bottled path costs $(C_0 + n c_A)/(n \rho)$ per unit quality. Setting them equal,

$$\frac{C_0 + n c_A}{n \rho} = c_L \implies n_\rho^* = \frac{C_0}{\rho c_L - c_A}.$$

Since $\rho < 1$, the denominator shrinks and $n_\rho^* > n^*$: pricing the quality loss raises the break-even workload. Bottling is the correct investment exactly when the expected workload size n exceeds n_ρ^* .

3.2 Quality-per-cost figures of merit

Two dimensionless ratios summarise a bottling outcome. The cost ratio is

$$r = \frac{c_A}{c_L},$$

and the quality-cost product — quality retained per unit of relative cost, equivalently the factor by which the bottled path multiplies macro-F1 delivered per unit cost — is

$$B = \frac{\rho}{r} = \rho \cdot \frac{c_L}{c_A}.$$

$B > 1$ means the bottled path dominates the direct path in quality-adjusted cost; B is the figure reported in Section 4 for the benchmark's headline successes. The reciprocal form, $1/B$ per unit of retained quality percentage, is an equivalent statement of the same quantity.

3.3 Statistical reanalysis of the reported counts

BOTTLED reports counts over $N = 60$ bottling runs (ten models, three tasks [1], [2]). Two counts are reanalysed.

Count 1: zero-shot confidence floor. Each run is compared against the lower bound of the 95% confidence interval of its own model's zero-shot performance. If a run's outcome were independent of its model's zero-shot distribution, the probability of falling below that floor by chance is the one-sided tail mass $p_0 = 0.025$ (the 95% two-sided interval leaves 2.5% in each tail). Under H_0 : 60 independent runs each below the floor with probability $p_0 = 0.025$, the count X is binomial:

$$X \sim \text{Binomial}(N = 60, p_0 = 0.025), \quad \mathbb{E}[X] = Np_0 = 1.5.$$

The reported observation is $X = 48$. Because $Np_0 = 1.5$ is small, the normal approximation is supplemented by an explicit log-arithmetic evaluation of the exact binomial tail,

$$P(X \geq 48) = \sum_{k=48}^{60} \binom{60}{k} (0.025)^k (0.975)^{60-k},$$

carried out in Section 4.2. A sensitivity variant with $p_0 = 0.05$ (allowing a one-sided interval construction) is also computed.

Count 2: distillation baseline. Each run is compared against the stronger of two small-model distillation baselines with the same token budget. Under the symmetric null that an agent's bottled artifact and the stronger baseline are equally good, each run loses with probability $p_0 = 0.5$:

$$X \sim \text{Binomial}(N = 60, p_0 = 0.5), \quad \mathbb{E}[X] = 30.$$

The reported observation is $X = 31$. Here $Np_0 = 30$, so the normal approximation is standard; the two-sided tail is computed in Section 4.3, and the one-sided upper tail is reported alongside the two-sided tail as the primary reading.

3.4 Proportion confidence intervals

For each observed proportion $\hat{p} = x/N$, the standard error is $\text{SE} = \sqrt{\hat{p}(1 - \hat{p})/N}$ and the 95% Wald interval is $\hat{p} \pm 1.96 \text{ SE}$. Wald intervals are used throughout; Wilson intervals are a standard alternative at these proportions, and no conclusion in this paper depends on the interval construction.

3.5 Scope and inputs

All input numbers come from the BOTTLED abstract as supplied in entries [1] and [2]: the counts 48 and 31 out of 60, the ten-models-three-tasks scope, the Opus 5 retention $\rho \approx 0.82$ at cost ratio $c_L/c_A \approx 657$, and the Jev comparison $\rho' \approx 0.94$ at one quarter of Jev's projected full-workload cost. No new data are collected, simulated, or assumed beyond explicitly labelled modelling assumptions. The setup cost C_0 is not reported in the supplied abstract; it is therefore carried symbolically, and all projections that require a numeric c_L or C_0 are labelled as projections with their assumptions stated.

4. Analysis

4.1 Observed proportions

Inputs (from [1], [2]): $x_1 = 48$ runs below the zero-shot 95% confidence floor, $x_2 = 31$ runs below the stronger distillation baseline, out of $N = 60$ runs.

$$\hat{p}_1 = \frac{48}{60} = 0.8, \quad \hat{p}_2 = \frac{31}{60} = 0.516667.$$

So 80% of bottling runs fall below the zero-shot confidence floor and approximately 51.7% fall below the distillation baseline.

4.2 Binomial null for the confidence-floor count

Inputs: $N = 60$, $p_0 = 0.025$, observed $x = 48$ (Section 3.3).

Expected count and standard deviation under H_0 :

$$\mu = Np_0 = 60 \times 0.025 = 1.5,$$

$$\sigma = \sqrt{Np_0(1-p_0)} = \sqrt{60 \times 0.025 \times 0.975} = \sqrt{1.4625} \approx 1.20934.$$

Normal-approximation z -score:

$$z = \frac{48 - 1.5}{1.20934} = \frac{46.5}{1.20934} \approx 38.45.$$

Exact tail evaluation. The single most likely value under H_0 is $k = 1$: with $\ln(0.975) \approx -0.025318$, $(0.975)^{59} = e^{59 \times (-0.025318)} = e^{-1.49376} \approx 0.22459$, so

$P(X = 1) \approx 60 \times 0.025 \times 0.22459 = 0.33689$ — the null predicts roughly one sub-floor run, not 48.

The dominant term of the upper tail is $P(X = 48) = \binom{60}{48} (0.025)^{48} (0.975)^{12}$. Since $\binom{60}{48} = \binom{60}{12}$, compute the binomial coefficient as the product $\prod_{i=1}^{12} \frac{48+i}{i} = \frac{49}{1} \cdot \frac{50}{2} \cdot \frac{51}{3} \cdot \frac{52}{4} \cdot \frac{53}{5} \cdot \frac{54}{6} \cdot \frac{55}{7} \cdot \frac{56}{8} \cdot \frac{57}{9} \cdot \frac{58}{10} \cdot \frac{59}{11} \cdot \frac{60}{12}$, which evaluates term-by-term to the integer

$$\binom{60}{12} = 1,399,353,840,075.$$

Then, with $\log_{10}(0.025) \approx -1.60206$ and $\log_{10}(0.975) \approx -0.01100$:

$$\begin{aligned} \log_{10} P(X=48) &= \log_{10}(1.39935 \times 10^{12}) + 48 \log_{10}(0.025) + 12 \log_{10}(0.975) \\ &= 12.14610 - 76.89888 - 0.13200 = -64.88478, \end{aligned}$$

so $P(X = 48) \approx 1.3 \times 10^{-65}$, and the full tail $P(X \geq 48)$ is of the same order because each subsequent term is smaller by a factor of at least $\frac{60-k}{k+1} \cdot \frac{0.025}{0.975} < 0.02$.

Sensitivity: even granting the null $p_0 = 0.05$ (one-sided interval construction), $\mu = 60 \times 0.05 = 3$, $\sigma = \sqrt{60 \times 0.05 \times 0.95} = \sqrt{2.85} \approx 1.68819$, and $z = (48 - 3)/1.68819 = 45/1.68819 \approx 26.66$. The observation is incompatible with "bottling preserves zero-shot performance" under either reading. The conclusion that strong zero-shot performance does not reliably translate into strong bottling capability [1], [2] is therefore not a marginal statistical artefact; it is an overwhelming rejection of the preservation null, and the failures are best read as a correlated, model-level property rather than per-run noise.

4.3 Binomial null for the distillation-baseline count

Inputs: $N = 60$, $p_0 = 0.5$, observed $x = 31$ (Section 3.3).

$$\mu = 60 \times 0.5 = 30, \quad \sigma = \sqrt{60 \times 0.5 \times 0.5} = \sqrt{15} \approx 3.87298.$$

$$z = \frac{31 - 30}{3.87298} \approx 0.25820.$$

The two-sided normal tail at $z = 0.258$ is approximately 0.80; the one-sided upper tail $P(X \geq 31)$ is approximately 0.40 (with continuity correction, approximately 0.45). Under either reading the count is far from any conventional significance level. The honest reading is that, as a population statistic, agent bottling performs like a coin flip against a fixed distillation baseline of equal budget: agents collectively add no detectable advantage over the simple baseline, even though individual runs (such as Opus 5's) can be strong.

4.4 Proportion confidence intervals

For $\hat{p}_1 = 0.8$:

$$SE_1 = \sqrt{\frac{0.8 \times 0.2}{60}} = \sqrt{\frac{0.16}{60}} = \sqrt{0.00266667} \approx 0.0516398,$$

$$0.8 \pm 1.96 \times 0.0516398 = 0.8 \pm 0.101214 \implies [0.6988, 0.9012].$$

For $\hat{p}_2 = 0.516667$:

$$SE_2 = \sqrt{\frac{0.516667 \times 0.483333}{60}} = \sqrt{\frac{0.249722}{60}} = \sqrt{0.00416204} \approx 0.0645140,$$

$$0.516667 \pm 1.96 \times 0.0645140 = 0.516667 \pm 0.126447 \implies [0.3902, 0.6431].$$

4.5 Quality-per-cost figures for the reported successes

Opus 5 bottled vs. its own zero-shot. Inputs from [1], [2]: $\rho = 0.82$, $c_L/c_A = 657$.

$$B_{\text{Opus}} = 0.82 \times 657 = 538.74.$$

Per unit of per-instance spend, the bottled artifact delivers about 539 times more macro-F1 than querying Opus 5 zero-shot, before accounting for the setup cost C_0 . Expressed reciprocally, the cost spent per retained percentage point of macro-F1 is

$$\frac{1}{B_{\text{Opus}}} = \frac{1}{657} \times \frac{100}{82} = 0.00152207 \times 1.219512 \approx 0.001856,$$

i.e., approximately 1.86×10^{-3} of the zero-shot per-instance cost per retained percent — the same quantity as $B = 538.74$, stated as its reciprocal.

The threshold retention for $B > 1$ at this cost ratio is

$$\rho_{\min} = \frac{1}{657} \approx 0.001522,$$

i.e., the bottled artifact only needs to retain about 0.15% of the zero-shot macro-F1 for the quality-adjusted cost ratio to favour bottling over direct querying, before setup costs. Opus 5's reported retention of 82% exceeds this threshold by a factor of $0.82/0.001522 \approx 539$.

Opus 5 bottled vs. Jev. Inputs from [1], [2]: $\rho' = 0.94$ (fraction of Jev's macro-F1 recovered) at one quarter of Jev's projected full-workload cost, so the relative cost ratio is $r' = 0.25$.

$$B_{\text{Jev}} = \frac{\rho'}{r'} = \frac{0.94}{0.25} = 3.76.$$

Per unit of spend, Opus 5's bottled artifact delivers approximately 3.76 times more macro-F1 than the purpose-built "system one" model Jev on the same task. The corresponding threshold retention is $\rho'_{\min} = r' = 0.25$: bottling must retain at least 25% of Jev's macro-F1 to be quality-adjusted cost-competitive, and the reported 94% exceeds this by a factor of $0.94/0.25 = 3.76$.

4.6 Break-even workloads

The raw break-even workload from Section 3.1, expressed in units of C_0/c_L , is

$$n^* = \frac{C_0}{c_L - c_A} = \frac{C_0/c_L}{1 - c_A/c_L} = \frac{C_0/c_L}{1 - 1/657} = \frac{C_0/c_L}{0.998478} \approx 1.00152 \frac{C_0}{c_L}.$$

Because the cost ratio is extreme ($c_A/c_L \approx 0.001522$), the raw break-even is essentially the setup cost measured in units of direct-query spend: the workload must be large enough that the saved per-instance spend repays C_0 .

The quality-adjusted break-even is

$$n_{\rho}^* = \frac{C_0}{\rho c_L - c_A} = \frac{C_0/c_L}{\rho - 1/657} = \frac{C_0/c_L}{0.82 - 0.001522} = \frac{C_0/c_L}{0.818478} \approx 1.2218 \frac{C_0}{c_L}.$$

Pricing the 18% quality loss raises the break-even workload by a factor $1.2218/1.00152 \approx 1.22$ relative to the raw break-even — a modest correction at this cost ratio, because the denominator is dominated by ρ rather than by the cost saving. The general sensitivity is: as $\rho \rightarrow 1/657$, $n_{\rho}^* \rightarrow \infty$; at $\rho = 0.5$, $n_{\rho}^* = \frac{C_0/c_L}{0.5 - 0.001522} \approx 2.006 C_0/c_L$; at $\rho = 0.25$, $n_{\rho}^* = \frac{C_0/c_L}{0.25 - 0.001522} \approx 4.025 C_0/c_L$. The break-even workload is therefore highly sensitive to retention when retention is low, and nearly insensitive to it when retention is high.

Against Jev, the same formula applies with $r' = 0.25$: $n_{\rho}^* = \frac{C_0/c_L}{\rho' - 0.25}$, so at the reported $\rho' = 0.94$ the break-even is $\frac{C_0/c_L}{0.69} \approx 1.449 C_0/c_L$ in units of Jev-equivalent spend.

4.7 Worked projection with stated assumptions

To make the break-even concrete, adopt two labelled assumptions (projections, not measurements): a direct per-instance cost of $c_L = 0.01$ dollars per query, and a setup cost of $C_0 = 100$ dollars (the token, compute and time budget of a bottling run, priced at API rates). Then:

- Raw break-even: $n^* = 100 / (0.01 \times 0.998478) \approx 10,015$ instances.
- Quality-adjusted break-even: $n_\rho^* = 100 / (0.01 \times 0.818478) \approx 12,218$ instances.
- At $n = 10^6$ instances, direct cost is $10^6 \times 0.01 = 10,000$ dollars; bottled cost is $100 + 10^6 \times 0.01 / 657 \approx 100 + 15.22 = 115.22$ dollars, a saving of approximately 9,884.78 dollars per million instances, delivered at 82% of the direct path's quality.

These figures scale linearly in both c_L and C_0 ; only the ratio C_0/c_L matters for the break-even, so a tenfold larger setup budget at the same c_L shifts the break-even to approximately 122,180 instances.

5. Results

The reanalysis yields four results, each computed in Section 4.

R1. The confidence-floor failure count is a structural property, not noise. Under the null that each run independently preserves its model's zero-shot performance, the expected number of sub-floor runs out of 60 is $\mu = 1.5$; the observed count is 48. The exact binomial tail is $P(X \geq 48) \approx 1.3 \times 10^{-65}$ (Section 4.2), with normal-approximation $z \approx 38.45$ ($z \approx 26.66$ under the one-sided-interval sensitivity at $p_0 = 0.05$). No plausible relaxation of the per-run independence assumption converts this into chance variation; the failures must be correlated across runs within models, i.e., a model-level bottling capability.

R2. The distillation-baseline count carries almost no evidence. The observed 31 of 60 runs below the stronger distillation baseline is consistent with a symmetric null: $z \approx 0.258$, two-sided $p \approx 0.80$, one-sided upper tail ≈ 0.40 (Section 4.3). Collectively, agent bottling performs like a coin flip against a fixed small-model baseline of equal token budget, even though individual runs can be strong.

R3. The reported successes are quality-adjusted cost-dominant by large factors. For Opus 5 on query-product relevance classification, $B_{\text{Opus}} = 0.82 \times 657 = 538.74$ against direct querying and $B_{\text{Jev}} = 0.94 / 0.25 = 3.76$ against Jev (Section 4.5). The minimum retention thresholds are $\rho_{\min} = 1/657 \approx 0.00152$ and $\rho'_{\min} = 0.25$ respectively.

R4. Break-even workloads are of order the setup cost measured in direct-query spend, inflated by roughly 22% for quality. $n^* \approx 1.00152 C_0/c_L$ and $n_\rho^* \approx 1.2218 C_0/c_L$ (Section 4.6). Under the labelled projection of Section 4.7 ($c_L = 0.01$ dollars, $C_0 = 100$ dollars), the quality-adjusted break-even is approximately 12,218 instances, and a million-instance workload saves approximately 9,884.78 dollars at 82% of the direct path's quality. *for the confidence – floor count and* [0.3902, 0.6431] for the baseline count (Section 4.4).

6. Discussion

The two headline counts behave in opposite ways under statistical reanalysis, and this asymmetry is the paper's central interpretive finding. The 48-of-60 count is so incompatible with independent per-run preservation that the only coherent readings are (a) bottling ability is a correlated, model-level property, or (b) the comparison itself is biased so that sub-floor outcomes are near-structural — for example if bottling inherently perturbs the task distribution enough that the zero-shot confidence interval is the wrong yardstick. Reading (b) would not rescue the preservation null; it would relocate the finding from "models fail at bottling" to "zero-shot calibration does not transfer to bottled deployment", which is operationally similar for a decision-maker: in neither case can zero-shot score be used to predict bottled outcome.

Limitations. The result rests on the following assumptions, with sensitivity noted for each:

- *Independence of runs under the null.* The 1.3×10^{-65} tail assumes 60 independent Bernoulli trials at $p_0 = 0.025$. If runs within a model are perfectly correlated (ten effective units), the exact tail is less extreme but the qualitative conclusion survives: even a worst case of all runs within each model sharing one outcome cannot produce 48 sub-floor runs unless at least eight of ten models are sub-floor by construction, which is itself the structural finding. The conclusion is robust; only the tail magnitude is sensitive.
- *The 2.5% tail reading of the confidence floor.* If the benchmark's intervals were one-sided, $p_0 = 0.05$ and $z \approx 26.66$ (Section 4.2); the conclusion is unchanged.
- *Reported summary statistics taken at face value.* The counts 48, 31, 60, the retention 0.82, the cost ratio 657, and the Jev figures are used exactly as the supplied abstract reports them; the abstract gives no confidence intervals on ρ or the cost ratio, and no detail on how "roughly 657 times lower reported cost" was computed. If the true cost ratio were, say, 100 rather than 657, then $B_{\text{Opus}} = 82$ and the qualitative dominance conclusion survives; only if the cost ratio fell below $1/0.82 \approx 1.22$ would the dominance flip.
- *Quality as a scalar.* The model compresses quality to macro-F1 retention ρ . Tasks where quality is multidimensional, or where errors are catastrophic rather than graded, are not represented; a bottled artifact at $\rho = 0.82$ may fail in correlated, adversarially exploitable ways that a 0.82 average conceals.
- *Setup cost carried symbolically.* C_0 is not reported in the supplied abstract; all numeric break-evens are projections conditional on the stated C_0 and c_L (Section 4.7), and scale linearly in C_0/c_L .
- *Single-source grounding.* Every empirical input comes from one benchmark abstract [1], [2]; no independent replication exists in the supplied bibliography, and the supplied summaries of the related works are in several cases truncated, limiting what can be drawn from them.

Falsification. The main result — that bottling failure is structural rather than per-run noise — would be falsified by any of the following observations: a reanalysis of BOTTLED's raw per-run data showing that the 48 sub-floor runs are concentrated in fewer models than eight of ten with within-model run-level variance comparable to zero-shot variance, which would make the independence assumption the wrong null; a published correction to the count or to the confidence-interval construction that raises the effective p_0 above approximately 0.4, at which point $z = (48 - 24)/\sqrt{60 \times 0.4 \times 0.6} \approx 6.32$ still rejects but the margin narrows to ordinary significance; or a replication on a different benchmark in which models with similar zero-shot scores bottling similarly, which would indicate that BOTTLED's spread measures benchmark-specific idiosyncrasy rather than a general capability axis. The quality-adjusted dominance result (R3) would be falsified by a corrected cost accounting that puts c_L/c_A below approximately 1.22 for the Opus 5 case.

Open questions. Three questions are not answerable from the supplied material. First, what determines the model-level bottling axis — architecture, training data, post-training regime — is unreported. Second, whether the 31-of-60 coin-flip result against the distillation baseline reflects genuine parity or a baseline that is itself near the achievable frontier at that token budget cannot be distinguished without per-task breakdowns. Third, the risk exposure of autonomously created artifacts under context confusion [4] is flagged but unquantified in the supplied summary, and the "Agentic Collapse" entry [13] supplies a title with no abstract, so no claim about systemic dynamics of agent-produced artifacts can be grounded here.

7. Practical Implications and Engineering Pathways

(a) Who would use the result and what changes. The immediate users are operators of high-volume, repetitive LLM workloads — classification, routing, labelling pipelines — and the platform teams who govern their spend. If the analysis holds, three decisions change. First, the workload-switch decision becomes an appraisal against $n_\rho^* \approx 1.22 C_0/c_L$: any workload whose expected volume exceeds roughly 1.22 times the setup cost measured in direct-query spend is a candidate, and the projection in Section 4.7 puts typical break-evens in the tens of thousands

of instances at current API price points. Second, model selection for a bottling agent cannot be made on zero-shot benchmark rank; the two-axis reading (zero-shot quality and bottling capability as separate axes, Section 4.2) implies that selection requires a bottling probe, and the LiveBench-grounded audit's rank-based landscape [10] is the kind of evidence that is insufficient for this choice. Third, procurement of agent-created artifacts acquires a governance step: before a workload is switched over, the artifact's provenance, validation budget and failure mode should be auditable, in line with the evaluation-and-governance bottleneck described in [11].

(b) Current readiness. This work is purely analytical, unpeer-reviewed, and built on a single benchmark's reported summary statistics; under the NASA Technology Readiness Level definitions it sits at TRL 1–2 (basic principles observed and reported; concept formulated). No artifact, prototype or measured deployment exists as part of this paper, and analytical work of this kind does not exceed TRL 3 even after validation against additional reported data. Claims of operational readiness for agent-created artifacts would be unsupported.

(c) The smallest next test. The cheapest de-risking step is a reanalysis, not an experiment: obtain BOTTLED's per-run results and test whether the 48 sub-floor runs cluster by model (a contingency-table test of run outcome against model identity). Clustering confirms the structural reading; a distribution consistent with independent per-run failures at an elevated constant rate would falsify the model-level interpretation and redirect practice toward per-run validation rather than model selection. The smallest experimental step beyond that is a single-task pilot: pick one workload above its projected n_ρ^* , run one bottling agent within a fixed budget, and measure realised ρ and c_A against the appraisal; a realised retention below $\rho_{\min} = 1/657$ against the direct path, or a realised cost ratio above 657 by more than the pilot's measurement uncertainty, refutes the transfer of R3 to that setting.

(d) Binding constraints. In order of likely first contact: (i) *validation data* — estimating ρ for a candidate artifact requires labelled instances, and the benchmark's workloads are unlabelled by construction, so a validation budget must be added to C_0 ; (ii) *quality tail risk* — a scalar retention conceals correlated error modes, and an artifact deployed on millions of instances amplifies any encoded bias or confusion [4], making monitoring the first operational cost; (iii) *setup-cost accounting* — C_0 is unreported in the supplied abstract, so every numeric break-even inherits the projection's assumption and must be re-priced per deployment; (iv) *governance and provenance* — no supplied entry describes a mechanism for machine-readable descriptions of agent-produced artifacts; the semantic-relationships mechanism [6] addresses protocol discovery among agents and is the nearest supplied prior art, but its extension to artifact provenance is an open question. Compute and energy constraints are not derivable from the supplied material and are likewise open.

(e) Order-of-magnitude bounds (projections). Under the assumptions of Section 4.7 ($c_L = 0.01$ dollars, $C_0 = 100$ dollars), the quality-adjusted break-even is approximately 1.2×10^4 instances, and the per-million-instance saving is approximately 9.9×10^3 dollars at $\rho = 0.82$. These scale linearly in C_0/c_L ; they are projections conditional on the stated assumptions, not measurements, and the underlying cost ratio 657 is the benchmark's reported figure for one task and one model.

(f) Risks and misuse paths. An autonomous bottling capability that works would let an agent choose, train and deploy the artifact that answers a production workload; an artifact built under context confusion encodes adversarial content into every downstream instance [4], and a value-misaligned artifact is a standing commitment made on a human's behalf, which is the alignment concern [8] frames for agent agreements. A governance process that audits provenance and validation before switchover, in the direction of [11], is the corresponding mitigation; its design is not solved here.

8. Conclusion

This paper reanalysed the BOTTLED benchmark's reported figures as an investment problem. The reanalysis separates two findings that the benchmark's abstract reports side by side but that behave oppositely under statistics: the 48-of-60 sub-floor count rejects the preservation null at approximately 1.3×10^{-65} under per-run independence and is best read as a correlated, model-level property, while the 31-of-60 count against the distillation baseline is statistically indistinguishable from a coin flip ($z \approx 0.26$, $p \approx 0.80$). On the reported successes, the quality-cost product reaches $B = 538.74$ against direct querying and $B = 3.76$ against the purpose-built Jev model, with quality-adjusted break-even workloads $n_\rho^* \approx 1.22 C_0/c_L$. The practical yield is a decision rule for workload switch-over, a warning that zero-shot rank cannot select a bottling agent, and a falsifiable null model — independent per-run preservation — against which future bottling results can be audited. The analysis is arithmetic on one benchmark's summary statistics, sits at TRL 1–2, and its next step is a clustering reanalysis of the per-run data; but the decision it formalises — when is a cheaper artifact worth its setup cost, and who audits the artifact — is one that every high-volume LLM operator already faces, with or without the theory.

Provenance and AI Involvement

- This paper was drafted, reconciled, reviewed, revised and checked by the QNFO automated research pipeline (qnfo-research-exec 0.16.0-revise-retry). Language models were tools in that process; they are not authors. The named author is accountable for the content.
- Models: drafting by openai/gpt-oss-120b, zai-org/glm-5.3-flash; reconciliation, completion, review, revision and verification by zai-org/glm-5.3-flash, zai-org/glm-5.3, openai/gpt-oss-120b, with the QNFO model router as a fallback.
- Sources: every reference is drawn from a bibliography retrieved from arXiv and the QNFO corpus before drafting; statements about a cited work are restricted to its retrieved title and abstract.
- Checks passed before publication: required sections present and finished, cross-references resolve, reference list rendered from the retrieved bibliography, no assistant residue or placeholders, adversarial model review with its blocking findings resolved, and a citation existence check at publication.
- Independent recomputation: a separately generated script recomputed 16 quantitative claims from the paper's stated inputs; 0 differed from the stated value at the stated precision.
- Not checked: whether each cited source supports the sentence that cites it beyond its abstract; derivation steps other than the recomputed claims. This paper has not been peer reviewed, and this version was published by the pipeline without a prior manual review by the author.

References

[1] TITLE: arXiv Query: `searchquery=&idlist=2610.08775&start=0&max_results=1` [2] Agent in a Bottle: Can LLM Agents Turn Their Capabilities Into Cheap, Scalable Artifacts?. arXiv:2610.08775v1. <https://arxiv.org/abs/2610.08775v1> [3] A Survey of Multi-Agent Deep Reinforcement Learning with Communication. arXiv:2203.08975v2. <https://arxiv.org/abs/2203.08975v2> [4] Agents at Risk: How Users Unwittingly Undermine LLM Safety. arXiv:2601.10758v3. <https://arxiv.org/abs/2601.10758v3> [5] Focus Agent: LLM-Powered Virtual Focus Group. arXiv:2409.01907v1. <https://arxiv.org/abs/2409.01907v1> [6] A mechanism for discovering semantic relationships among agent communication protocols. arXiv:2401.16216v1. <https://arxiv.org/abs/2401.16216v1> [7] Augmenting the action space with conventions to improve multi-agent cooperation in Hanabi. arXiv:2412.06333v3. <https://arxiv.org/abs/2412.06333v3> [8] Learning the Value Systems of Agents with Preference-based and Inverse Reinforcement Learning. arXiv:2602.04518v1.

<https://arxiv.org/abs/2602.04518v1> [9] A Plan Reuse Mechanism for LLM-Driven Agent. arXiv:2512.21309v2.
<https://arxiv.org/abs/2512.21309v2> [10] QNFO: Prioritizing Large Language Models for Scientific Research and
Agentic AI: A LiveBench-Grounded Audit (August 2026) [11] QNFO: Epistemic Legibility in AI-Assisted Science:
Ignorance Auditing as a Governance Instrument for the Peer-Review and Evaluation Bottleneck [12] QNFO: Flag
Structures and the Flagging Act: A Cross-Domain Survey with Explicit Counting Derivations and a Posed Re-Entry
Problem [13] QNFO: Agentic Collapse