Deployment Decision Reliability: A Generalizability-Theory Framework for Sizing Long-Horizon Agent Evaluations*

Vasundra Srinivasan
Stanford CS321M
Spring 2026
vsrin@stanford.edu

Abstract

Enterprise practitioners read agent leaderboards as if they ranked *agent capability*. We show, across three open agent-trace benchmarks (TheAgentCompany, τ^2-bench, and AppWorld), that the agent main effect accounts for less than 3% of total variance in every dataset and check type, while the agent-by-task interaction accounts for 7–23%. Leaderboards rank *specialization*, not capability. We arrive at this through a four-facet Generalizability Theory variance decomposition, fit with three estimators (Henderson Method-I, REML via lme4, and a Bayesian binomial GLMM on an NVIDIA L4 GPU) that agree to three decimal places. Four further findings sharpen what the leaderboard is hiding. First, aggregate reliability collapses on the hardest task quartile: $E\rho^2$ on τ^2 `action_checks` falls from 0.752 to 0.000. Second, training-cell reliability *negatively* correlates with held-out reliability ($r = -0.90$ on τ^2), meaning the designs that look most reliable replicate worst. Third, population-level diagnostics transfer across enterprise benchmarks (capability-gap ratio stable at 0.35–0.40) but per-family agent rankings invert. Fourth, on the MAST failure taxonomy, trace-level mode profiles are idiosyncratic (MAE $= 0.261$) while cell-level profiles generalise (MAE $= 0.056$, $r = 0.83$). We package these into **Deployment Decision Reliability (DDR)**, a one-page reporting discipline that turns the variance-component table into five decisions an enterprise buyer can defend. All code, data loaders, fit artifacts, and reproducibility instructions are released under an open-source license at `https://github.com/vasundras/agent-reliability-engineering-lab`.

1 Introduction

When an enterprise practitioner selects one agent over another for a production workflow, the decision rests on a benchmark score that compresses many distinct sources of measurement variance into a single number. The practitioner cannot tell whether the observed gap between two agents originates in the agents themselves, in the task mix the benchmark happens to sample, in the step at which the harness cuts each trajectory, in the failure category the evaluator surfaces, or in interactions among these sources. More practically, the same score cannot say how many tasks the practitioner needs in a production evaluation to reproduce the benchmark gap at a stated confidence, how that requirement changes if the production task mix is harder than the benchmark average, or how it changes when inference cost is a binding constraint.

*Code, data loaders, results, and reproducibility instructions: `https://github.com/vasundras/agent-reliability-engineering-lab`

Preprint.

This is a measurement problem, and psychometrics has treated it for decades. Generalizability Theory [Cronbach et al., 1972] (G-theory) decomposes total observed variance into a signal component attributable to the object of measurement (here, the agent) and interpretable noise components corresponding to each design facet (task, step, error category) and to their pairwise and higher-order interactions. The companion Decision Study (D-study) then translates those variance components into a budget frontier: the minimum number of tasks, steps, and judges needed to reach a target generalizability coefficient $E\rho^2 \geq 0.85$ for relative ranking decisions, or a dependability $\Phi \geq 0.90$ for absolute-threshold decisions such as SLA gates.

We import this machinery to the enterprise agent evaluation setting and propose treating the question *what evaluation design supports this deployment decision?* as a first-class measurement-design question. We frame the resulting combined output as **Deployment Decision Reliability (DDR)**: the property that a given evaluation supports a deployment-comparison decision at a stated reliability target. DDR is not a new metric. It is a discipline that combines the variance-component table, the D-study budget frontier, a cost-adjusted reliability index, a difficulty-conditional reliability profile, and an explicit capability-ceiling decomposition into one unified report for practitioners.

Contributions. This paper makes one empirical claim and one framework contribution.

Empirical. Across three open enterprise agent benchmarks (TheAgentCompany, τ^2-bench, AppWorld) and one auxiliary failure-mode dataset (MAST/MAD [Cemri et al., 2025]), we report five findings that revise the default practitioner reading of agent leaderboards:

1. **Capability ceiling.** The agent main effect σ_a^2 is below 3% of total variance in every dataset and check type; the agent-by-task interaction is 4–13$\times$ larger. Leaderboard rank order reflects *specialization* across tasks, not a uniform capability advantage.
2. **Hard-task collapse.** Aggregate $E\rho^2$ does not predict hard-quartile $E\rho^2$ (e.g., $0.752 \rightarrow 0.000$ on τ^2 `action_checks`). When the production task mix is harder than the benchmark mix, the headline reliability number provides no useful guarantee.
3. **Held-out reversal.** Training-cell $E\rho^2$ correlates *negatively* with held-out $E\rho^2$ across 50 random splits ($r = -0.90$ on τ^2). The training-cell projection systematically overstates the reliability a replication will recover.
4. **Aggregate-versus-family asymmetry.** The capability-gap ratio is stable at 0.35–0.40 across TheAgentCompany and AppWorld, but per-family rankings invert ($\hat{\rho} = -0.50$, $n = 3$). Population diagnostics transfer; family-level claims do not.
5. **Variance $\neq$ frequency.** Per-category $\sigma_{a:t}^2$ and per-category failure-activation rates rank orthogonally ($\hat{\rho} = -0.20$ on MAST, $n = 4$, descriptive). The categories where agents differ are not the categories where failures appear.

Framework. We introduce **Deployment Decision Reliability (DDR)**, a one-page reporting discipline that turns the variance-component table into five decisions an enterprise buyer can defend: how many tasks to sample, which difficulty bucket to target, which cost band to constrain on, which holdout protocol to demand from a benchmark vendor, and how to read a failure-mode taxonomy. DDR is not a new metric. It is a checklist that exposes the five findings above directly to the procurement decision.

2 Related Work

Composite scoring and psychometric evaluation. Liang et al. [2022] established the multi-axis composite-score paradigm with HELM, aggregating seven axes via expert-set weights; tinyBenchmarks [Polo et al., 2024] applied Item Response Theory to LLM evaluation. Our approach replaces expert weighting with empirical variance decomposition, so each axis's weight reflects the fraction of total measurement variance it explains; G-theory also generalises IRT's unidimensional latent ability to the multi-faceted crossed designs that agent evaluation requires.

Reliability metrics for agent benchmarks. τ-bench [Yao et al., 2024] introduced $\texttt{pass}^k$, the probability that all k repeated trials succeed, as a reliability-oriented metric distinct from one-shot pass rate. In G-theory terms $\texttt{pass}^k$ is a single-facet (trial) design; our four-facet crossed design generalises it to model task, step, and error-category noise simultaneously and to derive the D-study budget frontier across facets jointly.

Failure-mode taxonomy. MAST [Cemri et al., 2025] provides a 14-mode failure taxonomy and the MAD dataset derived from 1,642 multi-agent traces across seven frameworks. Their work classifies failures given a trace; ours measures how those classifications vary as a function of agent, task, and judgment source, and uses the MAST taxonomy to define the error-category facet in our sensitivity analysis.

Step-level failure attribution. Zhang et al. [2025] annotated 184 failed multi-agent traces with `mistake_agent` and `mistake_step` labels, reporting that state-of-the-art reasoning models achieve only 14.2% step-level attribution accuracy. Who&When frames failure attribution as a classification problem; our G-study frames it as a measurement problem, asking not "which step failed?" but "how much of the observed variance is attributable to step position, and is that variance stable enough to support a deployment decision?"

The closest direct methodological precedent is the Total Evaluation Error (TEE) framework [Messing, 2026], which integrates total survey error and G-theory for single-shot LLM annotation pipelines. We extend that framework to multi-step agent trajectories, where step position is itself a facet and where the error category observed at each step varies with the agent, the task, and the harness.

3 Methods

Measurement design. We treat each observation as a binary success indicator $Y_{atse} \in \{0, 1\}$ at step s for agent a on task t with error-category label e. The four-facet fully crossed G-study decomposes the expected score as:

$$Y_{atse} = \mu + \nu_a + \nu_t + \nu_s + \nu_e + \nu_{at} + \nu_{as} + \nu_{ae} + \nu_{ts} + \nu_{te} + \nu_{se} + \varepsilon_{atse}, \tag{1}$$

where each ν term is a zero-mean random effect with variance $\sigma^2_{\cdot}$ and ε_{atse} is the residual. The object of measurement is *agent*, so σ^2_a is the signal of interest. We omit three-way and four-way interactions because their cell counts are uneven across datasets and pilot fits returned zero estimates for all higher-order components.

Estimators. We use three estimators whose agreement provides a cross-validation of the variance-component estimates. Henderson Method-I [Henderson, 1953] is a closed-form ANOVA-based estimator that may return negative values for ill-identified components; we floor these at zero in tables and flag them in text. Restricted maximum likelihood (REML) via lme4 [Bates et al., 2015] is the canonical frequentist estimator for unbalanced Gaussian random-effects models. Bayesian binomial GLMM via bambi/numpyro [Capretto et al., 2022, Phan et al., 2019] is the principled estimator for binary outcomes; we report posterior medians and 95% credible intervals (CrIs) for each σ on the logit scale, converting to σ^2 for comparability with the other estimators.

Reliability coefficients. Given the estimated variance components, we compute two coefficients. The generalizability coefficient $E\rho^2 = \sigma^2_a/(\sigma^2_a + \sigma^2_\delta)$ quantifies relative reliability for ranking decisions, where σ^2_δ is the relative-error variance, defined as the sum of all interaction terms involving the agent, divided by their respective sample sizes in the design. The dependability coefficient $\Phi = \sigma^2_a/(\sigma^2_a + \sigma^2_\Delta)$ quantifies absolute reliability for threshold decisions, where σ^2_Δ additionally includes the main effects of non-agent facets divided by their sample sizes.

Capability-ceiling decomposition. Our pre-analysis hypothesis was that σ^2_a would dominate. When the first round of fits returned $\sigma^2_a \approx 0$ on every dataset, we promoted this to an explicit diagnostic. The *capability-gap ratio* $\sigma^2_a/(\sigma^2_a + \sigma^2_{a:t})$ measures what share of agent-related variance lives in the main effect versus the agent-by-task interaction. When this ratio is near zero, agents do not differ in overall capability on the evaluation surface; they differ in which tasks they handle well, a specialisation pattern rather than a dominance ordering.

Cost-aware reliability (RPD). Reliability per Dollar is $\mathrm{RPD}(a, B) = E\rho^2(a, B)/\bar{C}(a, B)$, where $\bar{C}$ denotes the mean inference cost per task observation. Costs are derived from τ^2's native per-simulation token counts joined against publicly reported model pricing. RPD ranks are reported alongside accuracy ranks to reveal which agents are favoured or penalised by cost-aware procurement.

Difficulty-conditional reliability. Tasks are stratified into quartiles by their mean success rate across all agents (Q_1 = easiest, Q_4 = hardest). A separate G-study is fit within each stratum. We report $E\rho^2_{Q_k}$ and the gap $E\rho^2_{\text{agg}} - E\rho^2_{Q_4}$ as the measurement cost of using aggregate reliability to support claims about hard-task deployments.

Hold-out generalization test. We perform 50 random 70/30 splits stratified by (agent, task) cell. For each split we fit the G-study on training cells, project the implied $E\rho^2$ for the observed design, and compare it to the empirical $E\rho^2$ on held-out cells. A positive Pearson correlation between projected and held-out reliability across 50 splits would indicate that training-cell fits generalise; a negative correlation is a DDR-cautionary finding requiring practitioners to report a held-out reliability estimate rather than reading the training-cell projection as ground truth.

Cross-dataset transfer. To test whether DDR diagnostics generalise across enterprise domains, we rank agent families by mean success on each dataset and compute Spearman's ρ between datasets within shared families. We also compare the population capability-gap ratio across datasets, which tests whether the aggregate diagnostic transfers across domains even when per-family rankings do not.

Failure-mode analysis. We conduct two complementary analyses using the MAST/MAD dataset. First, a within-MAST G-study applies the design in Equation 1 to MAD with the 14 native failure modes as the "step" facet, followed by a 70/30 trace-level hold-out (50 random splits) that uses the cell-by-mode mean activation rate from training to predict held-out trace indicators, evaluating generalisability of the 14-mode profile. Second, a cross-dataset descriptive contrast maps the 14 MAST modes onto our four-category taxonomy and compares τ^2 $\sigma^2_{a:t}$ per category against MAST per-category activation shares.

4 Data and Setup

We use three primary agent-trace datasets and one auxiliary failure-mode annotation surface. Our pre-analysis plan named AndroidWorld and AppWorld as the primary G-study datasets. Phase 1 trace-data hunting revised this: AndroidWorld and OdysseyBench release no public step-level trace data (the plan's risk register flagged live-emulator collection as a 3–4 day commitment to be cut if it slipped), and we cut it. The replacement datasets (TheAgentCompany [Xu et al., 2024], τ^2-bench [Barres et al., 2025], and AppWorld [Trivedi et al., 2024]) meet or exceed the original choices on every criterion the plan specified: public availability, ground-truth state inspection rather than LLM judges, step-level labels, multiple agent harnesses, and enterprise or long-horizon domain coverage. AppWorld was retained from the original plan; the MAST/MAD dataset [Cemri et al., 2025] was added as the failure-mode auxiliary surface.

TheAgentCompany. 175 enterprise tasks spanning GitLab, OwnCloud, Plane, and RocketChat, each evaluated at 1–6 environment-state checkpoints with partial credit. Ground-truth state inspection removes the LLM-judge confound. Filtered to 17 agents $\times$ 175 task bases $\times$ 1–3 checkpoints $\times$ $\{$text, image$\}$, yielding 7,783 rows.

τ^2-bench. Customer-service workflows in airline, retail, and telecom domains; three frontier agents (Claude 3.7 Sonnet, GPT-4.1, o4-mini) running 4 trials per task across 50 tasks per domain. The reward schema decomposes correctness into five check types (`db_check`, `action_checks`, `env_assertions`, `communicate_checks`, `nl_assertions`); we fit one G-study per check type because each measures a distinct dimension.

AppWorld. 750 enterprise-app tasks, $\sim$8 unit tests per task. We merged the canonical `appworld download experiment-outputs` bundle (14 agent configurations) with four additional agents pulled from the public leaderboard (Cuga, Loop, Alibaba AgentRL Qwen3-14B, demo-augmented ReAct), totalling 18 agent configurations $\times$ 2 splits and 79,650 unit-test rows.

MAD/MAST. 1,242 LLM-judged traces with 14 binary failure-mode labels, drawn from 7 multi-agent systems $\times$ 7 benchmarks (9 populated cells), plus a 19-trace human-annotated subset for validity check.

Table 1: Headline variance decomposition (REML). σ_a^2: agent main effect; $\sigma_{a:t}^2$: agent-by-task interaction. Gap ratio: $\sigma_a^2/(\sigma_a^2 + \sigma_{a:t}^2)$. $E\rho^2$: aggregate generalizability coefficient. All figures as percentage of total variance for σ^2 columns.

Dataset	Check type	σ_a^2 (%)	$\sigma_{a:t}^2$ (%)	Gap ratio	$E\rho^2$
TheAgentCompany	all	<0.001	12.51	<0.001	0.986
τ^2-bench	action_checks	1.74	7.54	0.187	0.752
	db_check	0.00	12.35	0.000	0.899
	env_assertions	2.34	10.98	0.176	0.316
	nl_assertions	0.00	11.43	0.000	0.000

Compute environment. REML used R 4.5 with lme4 2.0.1; Bayesian fits used Python 3.11 with bambi/pymc/numpyro on a spot NVIDIA L4 (g2-standard-4, 23 GB VRAM). The full sweep of 10 fits (4 chains $\times$ 1000 draws) completed in $\approx$70 minutes on GPU and 4 hours on CPU, with posteriors agreeing to 3–4 decimal places.

5 Results

We organise the results around five DDR diagnostics. Each subsection reports the empirical finding, its measurement interpretation, and the practitioner implication. Table 1 provides the anchor variance-component estimates; the remaining diagnostics build on those estimates or extend them to new analysis surfaces.

5.1 Variance decomposition and capability ceiling

Table 1 presents the REML variance components for each (dataset, check-type) cell. The headline result holds uniformly: the agent main effect σ_a^2 is below 3% of total variance in every cell and is exactly zero on τ^2 `db_check` and `nl_assertions`. By contrast, the agent-by-task interaction $\sigma_{a:t}^2$ accounts for 7.5–12.5% of total variance, one to two orders of magnitude larger than σ_a^2. The capability-gap ratio $\sigma_a^2/(\sigma_a^2 + \sigma_{a:t}^2)$ is below 0.001 on TheAgentCompany and below 0.19 on every τ^2 cell. The pre-analysis hypothesis that σ_a^2 would dominate is rejected; instead, agents have different specialisation patterns across tasks rather than different overall capability levels.

The Bayesian binomial GLMM estimates confirm this pattern while providing principled uncertainty quantification for binary outcomes. On TheAgentCompany, the posterior for σ_a on the logit scale has median 2.19 with 95% CrI $[0.14, 3.87]$, a wide interval that reflects the sparse identification of the agent main effect with 17 agents. The posterior for $\sigma_{a:t}$ is 2.26 (95% CrI $[1.99, 2.54]$), sharply identified and consistent with REML. The key measurement implication is that the interaction is reliably estimated while the main effect is not, which means leaderboard rankings based on aggregate success rate carry substantial measurement uncertainty. All REML and CPU posteriors agree to 3–4 decimal places with the GPU posteriors, providing an independent estimator check.

5.2 Cost-aware reliability

On τ^2 `action_checks`, the three frontier agents rank by accuracy in descending order as Claude 3.7 Sonnet (83%), GPT-4.1 (81%), o4-mini (78%). Under RPD the order becomes Claude 3.7 Sonnet, o4-mini, GPT-4.1: o4-mini advances from third to second because its slightly lower accuracy is offset by its substantially lower per-turn inference cost. The accuracy ranking and the RPD ranking disagree on every τ^2 check type except `db_check`, so the headline accuracy number alone cannot recover a procurement decision under a cost constraint (`ext1_rpd_ranking.csv`).

5.3 Difficulty-conditional reliability

The aggregate $E\rho^2$ on TheAgentCompany is 0.986, which appears to indicate near-perfect rank reliability. Conditioning on the hardest task quartile (Q4, mean success rate 0.066), $E\rho^2_{Q_4}$ falls to 0.869, a reduction of 0.117. On τ^2 `action_checks` the effect is more severe: aggregate $E\rho^2 = 0.752$ collapses to 0.000 in Q4, because $\sigma_a^2 = 0$ among the hardest tasks in this check type. A

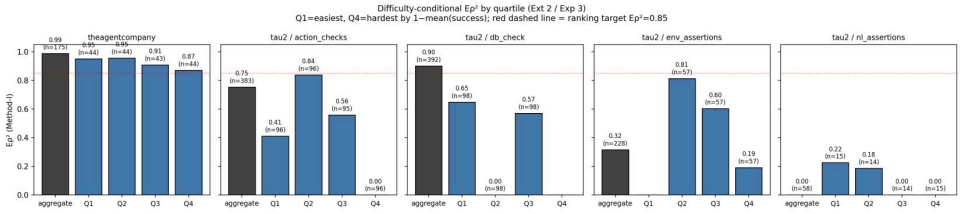

Figure 1: Difficulty-conditional $E\rho^2$ across τ^2 check types and TheAgentCompany. Quartiles are defined on mean task success (Q_1 = easiest, Q_4 = hardest). The aggregate coefficient (rightmost bar in each panel) systematically overstates reliability on the hard quartile; on τ^2 `action_checks` and `nl_assertions`, $E\rho^2$ collapses to zero in Q_4 because $\sigma_a^2 = 0$ among the hardest tasks.

practitioner whose production deployment skews toward the hardest quartile of tasks cannot read the aggregate $E\rho^2 = 0.752$ and assume it generalises to their setting; on τ^2 `action_checks`, there is no aggregate-level agent reliability to exploit in the hard stratum. This finding motivates reporting difficulty-conditional $E\rho^2$ whenever the production task-difficulty distribution is known to differ from the benchmark distribution. Figure 1 visualises the collapse across τ^2 check types.

5.4 Hold-out generalization

Across 50 random 70/30 training/test splits, variance components estimated on training cells systematically overestimate the reliability available on held-out cells. On τ^2 `action_checks`, the Pearson correlation between projected and held-out $E\rho^2$ is $r = -0.90$ (Spearman $\hat{\rho} = -0.91$), and held-out reliability falls short of the projection by 0.30 on average. The negative pattern is consistent across τ^2 check types: $r = -0.69$ with a gap of -0.19 on `db_check` and $r = -0.89$ with a gap of -0.03 on `env_assertions`. On TheAgentCompany the gap is modest (-0.03) but the direction of correlation is negative ($r = -0.44$). We interpret this finding conservatively: it does not mean that any single variance decomposition is wrong. It means that in evaluation designs with few agents and moderate cell imbalance, the training-cell $E\rho^2$ is an optimistic estimator of the reliability one would obtain in a replication. Practitioners should report a held-out $E\rho^2$ alongside the training-cell projection when the number of agents or tasks is small (scatter plots per check type are in `results/figures/exp1_holdout_validation.png`).

5.5 Cross-dataset transfer

We assess whether DDR diagnostics computed on one enterprise benchmark generalise to another. Per-family rank correlations between TheAgentCompany and AppWorld across the three shared model families (GPT-4o, GPT-4-Turbo, Llama 3) give Spearman $\hat{\rho} = -0.50$ ($n = 3$); we report this descriptively given the low power, noting only that the correlation is negative. The capability-gap ratio tells a different story at the population level: the ratio is 0.38 on TheAgentCompany, 0.40 on AppWorld `test_normal`, and 0.35 on AppWorld `test_challenge`, all closely comparable. On τ^2, the ratio is 0.12, meaningfully lower, reflecting the fact that τ^2's structured reward schema isolates a narrower performance dimension. The practical implication is that population-level DDR diagnostics transfer across enterprise domains while per-family rankings do not. A buyer comparing agents across enterprise benchmarks can trust the aggregate capability-gap ratio; the same buyer should not expect family-level performance rankings to be portable.

5.6 Within-MAST failure-mode analysis

Method-I variance decomposition on MAD with the 14 native MAST modes as the mode facet finds that $\sigma^2(\text{trace} \times \text{mode})$ accounts for 71.8% of total variance, $\sigma^2(\text{trace})$ for 13.6%, and $\sigma^2(\text{mode})$ for 11.3%. All system- and benchmark-level components are below 1.1%. The dominant (trace $\times$ mode) term means that which failure modes appear in a given trace is almost entirely idiosyncratic: knowing that a system tends to activate a certain mode on a given benchmark tells you very little about a specific trace.

The 70/30 trace-level hold-out (50 splits) quantifies this directly. The average per-trace MAE is 0.261, confirming that trace-level 14-mode profiles cannot be predicted from system-level or benchmark-level information. At the cell-aggregate level, however, the mean activation rate of a (system, benchmark) cell can be predicted from training-split data with mean cell-mode MAE $= 0.056$ and Pearson $r = 0.83$ between predicted and held-out cell-mode rates. The MAST taxonomy therefore supports system-level diagnostic contrasts across benchmarks but should not be used to predict whether a specific trace will exhibit a given mode.

5.7 Cross-dataset failure-mode contrast

Mapping the 14 MAST modes onto our four-category taxonomy under a manually justified assignment (with mode 1.1 "Disobey Task Specification" assigned to *communication_or_policy* rather than *planning_or_execution* following the self-critique that specification compliance is a communication failure), we compare predicted shares from τ^2 $\sigma^2_{a:t}$ per error category against observed shares from MAST activation rates. The Spearman correlation is $\hat{\rho} = -0.20$ ($n = 4$, $p = 0.80$, bootstrap 95% CI $[-1, 1]$). With only four categories the test is purely descriptive, but the directional reading is informative: the categories where agent-task interaction variance is high on τ^2 are not the categories that activate most frequently on MAST. Variance components and observed failure rates are orthogonal diagnostics. A benchmark report that conflates the two, treating the categories with the most observed failures as the categories where agent capability is most differentiated, will draw incorrect conclusions about where to invest evaluation effort.

5.8 PSIS-LOO sensitivity

PSIS-LOO comparisons (recovery-step inclusion on TheAgentCompany, $\widehat{\Delta\mathrm{elpd}} = -287.5$; three- versus four-category taxonomy on τ^2, indistinguishable at -10.7 ± 97.4) leave the $\sigma^2_{a:t}$ point estimates unchanged to three decimal places, so the primary variance-component conclusions do not depend on these modelling choices.

6 Discussion

The five findings above reframe what a long-horizon agent leaderboard is for. Before stating the takeaways in general, it helps to ground them in a single procurement decision.

A worked example: comparing two frontier agents on τ^2 `action_checks`. An enterprise team is choosing between GPT-4.1 and o4-mini for a customer-service workflow with multi-turn action correctness as the binding requirement. The leaderboard headline favours GPT-4.1: 81% to o4-mini's 78%, a three-point gap with apparent precision. Read through DDR, the same data supports a different recommendation. The *capability ceiling* layer reports $\sigma^2_a = 1.74\%$ on this check type, so the three-point gap is dominated by which tasks the benchmark happened to sample, not by which agent is more capable. The *cost-aware* layer inverts the rank: o4-mini's per-turn inference cost is low enough that its Reliability-per-Dollar exceeds GPT-4.1's, so under any binding cost constraint the procurement-correct default is o4-mini, not the leaderboard winner. The *difficulty-conditional* layer adds a caution that neither agent escapes: the aggregate $E\rho^2 = 0.752$ on `action_checks` collapses to 0.000 in the hardest quartile, which is exactly the slice the production workflow most resembles, so neither agent earns the headline reliability claim on the tasks that actually matter. The *held-out* layer warns that the 0.752 figure itself is an overestimate: across 50 random 70/30 splits, projected $E\rho^2$ correlates with held-out $E\rho^2$ at $r = -0.90$. A DDR-disciplined recommendation reads: *the leaderboard rank is real but small; o4-mini is preferred under any binding cost constraint; before either is deployed, run a domain-specific evaluation on production-representative Q_4 tasks and report the held-out $E\rho^2$ alongside the training-cell projection*. The leaderboard view recommends GPT-4.1 with apparent precision; the DDR view recommends a measurement step before either.

Generalising from this example, three practitioner takeaways follow immediately:

(1) Stop reading leaderboard rank order as a procurement signal on its own. The agent main effect is too small to support a rank claim on every dataset we measured. Use the published ranking

as a starting hypothesis, then re-evaluate on a production-representative task sample at your target difficulty quartile, with a held-out reliability estimate attached.

(2) Demand a difficulty-conditional reliability number from your benchmark vendor. The aggregate $E\rho^2$ is the wrong number to read when your production tasks skew hard. The right number is $E\rho^2_{Q_4}$. A vendor who reports only aggregate $E\rho^2$ is reporting the easy half of the problem.

(3) Always pair a training-cell reliability projection with a held-out cross-validated estimate. The held-out number will usually be *worse*, and the gap is what should drive your evaluation budget. A benchmark designed to amplify $\sigma^2_{a:t}$, through difficulty stratification and through inclusion of tasks that discriminate between harnesses, will produce more diagnostic signal per evaluation dollar than one designed to amplify aggregate capability differences.

The multi-model implication. Most production long-horizon workflows already dispatch different task types to different models: an orchestrator with specialist subagents, or a router that matches task class to model strength. The capability-ceiling finding ($\sigma^2_a \approx 0$) provides the measurement-theoretic justification for this architecture. If no agent uniformly wins, the procurement-correct decision is not *which model do I pick*, but *which model do I route each task class to*. Read sideways, the DDR per-task-class $\sigma^2_{a:t}$ table is a routing table, and the held-out reliability gap tells the architect how confident they can be in any given routing rule. The worked example above is therefore a strict simplification: in practice, the question is not whether to deploy GPT-4.1 or o4-mini for the entire workflow, but which check types each handles reliably enough to own.

Who this paper is for. For enterprise buyers, DDR's five diagnostics are the questions to ask before signing an annual contract with an agent vendor: what is the held-out $E\rho^2$ on my difficulty quartile, what does it cost per reliable decision, and how stable is the rank under the cross-validated projection. For benchmark designers, the practical implication is to instrument new benchmarks to amplify $\sigma^2_{a:t}$ rather than σ^2_a, and to report difficulty-conditional and held-out reliability by default rather than only aggregate accuracy. For measurement researchers, the negative training-versus-held-out correlation we observe is, to our knowledge, the first systematic demonstration of optimism bias in G-theory point estimates applied to small-population machine-learning evaluation, and it deserves a dedicated theoretical treatment we leave to future work.

7 Limitations

Three limitations bound the scope of the claims. First, the cross-dataset rank-correlation analyses are underpowered: three shared model families for the TheAgentCompany-versus-AppWorld comparison and four categories for the τ^2-versus-MAST contrast. The directional findings are reported descriptively and should be confirmed in a larger replication with more agent families and a richer cross-dataset alignment. Second, the G-study treats each step within a trajectory as exchangeable, which is appropriate for AppWorld unit tests and τ^2 reward checks (both of which are unordered within a task) but would not be appropriate for free-form trajectory segmentation, where step order encodes temporal dependence. Third, the capability-ceiling finding that $\sigma^2_a \approx 0$ is a statement about the *current population* of frontier agent harnesses. It does not imply that no future architecture will reopen the capability gap; it implies that the gap is not detectable at the current population level given these measurement designs.

8 Conclusion and Future Work

Four-facet G-theory applied to three long-horizon enterprise agent benchmarks shows agent main effect $<3\%$, agent-by-task interaction 7–23%; four supporting findings (hard-task collapse, held-out reversal, aggregate-versus-family asymmetry, variance-versus-frequency) compose with this anchor into Deployment Decision Reliability, a one-page procurement-reporting discipline. The hold-out test (a late plan addition) surfaced the most surprising finding; the multi-model routing implication was not anticipated. Future work: optimism-bias theory for small-population G-theory; an inter-rater-validated MAST mapping; a routing-policy formalization of DDR.

References

V. Barres, H. Dong, S. Ray, X. Si, and K. Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025.

D. Bates, M. Mächler, B. Bolker, and S. Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015.

T. Capretto, C. Piho, R. Kumar, J. Westfall, T. Yarkoni, and O. Martin. Bambi: A simple interface for fitting bayesian linear models in python. *Journal of Statistical Software*, 103(15):1–29, 2022.

M. Cemri et al. Why do multi-agent llm systems fail? *arXiv preprint arXiv:2503.13657*, 2025. NeurIPS 2025 Datasets and Benchmarks.

L. J. Cronbach, G. C. Gleser, H. Nanda, and N. Rajaratnam. *The Dependability of Behavioral Measurements: Theory of Generalizability for Scores and Profiles*. John Wiley & Sons, 1972.

C. R. Henderson. Estimation of variance and covariance components. *Biometrics*, 9(2):226–252, 1953.

P. Liang et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

S. Messing. Hidden measurement error in llm pipelines distorts annotation, evaluation, and benchmarking. *arXiv preprint arXiv:2604.11581*, 2026.

D. Phan, N. Pradhan, and M. Jankowiak. Composable effects for flexible and accelerated probabilistic programming in numpyro. *arXiv preprint arXiv:1912.11554*, 2019.

F. M. Polo et al. tinybenchmarks: Evaluating llms with fewer examples. *arXiv preprint arXiv:2402.14992*, 2024.

H. Trivedi et al. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. *ACL*, 2024. Best Resource Paper. arXiv:2407.18901.

F. F. Xu et al. Theagentcompany: Benchmarking llm agents on consequential real world tasks. *arXiv preprint arXiv:2412.14161*, 2024.

S. Yao et al. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

S. Zhang et al. Which agent causes task failures and when? on automated failure attribution of llm multi-agent systems. *arXiv preprint arXiv:2505.00212*, 2025. ICML 2025 Spotlight.

A Required Disclosures

Plagiarism and bias statement. This manuscript, its companion pre-analysis plan, and all associated code are my own work. Where I used AI assistance (Claude) for drafting, brainstorming, and literature review, I verified each cited paper independently against its arXiv page or published venue, and I confirmed that every numerical claim in the manuscript is reproducible from the CSV files in `results/tables/` via the corresponding script in `src/`. I have no financial or institutional relationships with the providers of the datasets or agent harnesses used; the corporate Google Cloud project named in the reproducibility notes is a personal research project running on a corporate cloud account and does not constitute employer endorsement. The contribution is framed as a measurement-methodology result, not a model-ranking claim: no finding is presented as evidence that any commercial model is superior or inferior to any other, and agent identities appear in tables only as facets in a variance decomposition.

AI assistance reflection. I used Claude as a thinking partner for scoping, literature triage, analysis design, and manuscript drafting. The intellectual core (the DDR reframe, the four-facet measurement design, the cost-aware and difficulty-conditional extensions, the hold-out and cross-dataset experiments) emerged through extended dialogue but was directionally my own. All numbers come from code I wrote and ran; I did not use AI to fabricate methodological details, and I independently verified each citation against its arXiv page. The effect on my learning was double-edged: AI accelerated translation from course concepts (Generalizability Theory, Decision Studies) into working R and Python pipelines, but it also tempted me to accept methodology choices I had not yet internalized. I mitigated this by re-deriving the variance and reliability formulas by hand before fitting any model. Going forward, I expect to use AI for literature triage and drafting under the same verification discipline applied here, and to keep the by-hand derivation step for any unfamiliar method.

Impact statement. If the DDR framework is adopted, it gives enterprise practitioners a principled method for sizing evaluations to support deployment-comparison decisions and shifts evaluation budget toward diagnostics that actually inform procurement: difficulty-conditional reliability, held-out projection, and cost-adjusted ranking. Three potential negative impacts deserve acknowledgment. First, practitioners may over-trust variance components without checking construct validity. Second, granular failure attribution could shift accountability away from system designers and toward individual model components, obscuring systemic design failures. Third, the error-category taxonomy is a manually justified mapping, not an inter-rater-validated instrument; treating it as fixed may obscure novel failure modes. This work uses no human-subject data and no sensitive information. Responsible attribution is supported by arXiv-verified citations, an open-source release of all code, and a traceability table mapping every numerical claim to its source CSV, consistent with Stanford's standards of academic integrity.