NBER WORKING PAPER SERIES

ARTIFICIAL JAGGED INTELLIGENCE: WHEN AI BENCHMARKS MISSTATE DEPLOYMENT VALUE

Joshua S. Gans

Working Paper 34712
http://www.nber.org/papers/w34712

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
January 2026, Revised June 2026

This paper replaces an earlier version of the paper called “A Model of Artificial Jagged Intelligence." Thanks to Tom Cunningham for helpful comments. Research assistance from Refine.ink, Claude Opus 4.8, and ChatGPT 5.5 are acknowledged. Funding from the SSHRC is gratefully acknowledged. Responsibility for all errors remains our own. The views expressed herein are those of the author and do not necessarily reflect the views of the National Bureau of Economic Research.

The author has disclosed additional relationships of potential relevance for this research. Further information is available online at http://www.nber.org/papers/w34712

Artificial Jagged Intelligence: When AI Benchmarks Misstate Deployment Value
Joshua S. Gans
NBER Working Paper No. 34712
January 2026, Revised June 2026
JEL No. D83, O33

ABSTRACT

Organisations increasingly select and deploy artificial intelligence systems on the strength of public benchmarks. A benchmark, however, scores a system on a single distribution of tasks, whereas each organisation meets its own. Because AI performance is uneven across tasks, a property called artificial jagged intelligence, these distributions diverge, and a system that looks reliable on average can fail on the tasks a given workflow uses most. We model this gap and show that it is not noise but a predictable exposure effect: deployment loss exceeds benchmark loss exactly when the tasks an organisation uses most are those the system handles worst. This single mechanism links managerial choices usually studied in isolation. It governs when to roll out a system, where to direct scarce reliability investment, whether to audit one's own task mix before committing, and when to verify outputs after deployment. Better information about the workflow redirects investment towards targeted fixes whose value a public benchmark hides. The same logic explains why a single benchmark score is not enough: providers should report performance by task category so that organisations can reweight it for their own use.

Joshua S. Gans
University of Toronto
Rotman School of Management
and NBER
joshua.gans@rotman.utoronto.ca

1 Introduction

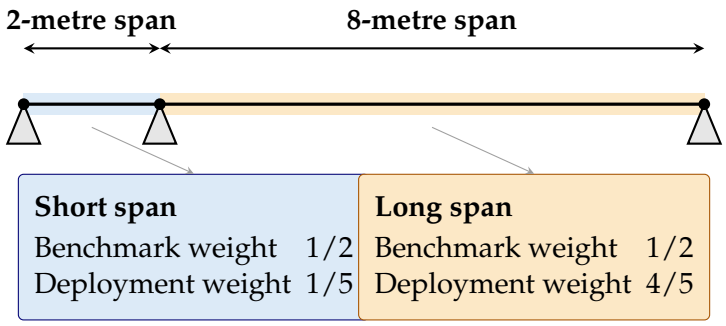
An organisation evaluating an artificial intelligence system will often begin with a benchmark. A legal team might test contract summaries, a customer-service unit might score suggested replies, or a software group might review generated code. The resulting average is useful. It is also incomplete. The tasks included in an evaluation need not appear with the same frequency after rollout. A system can perform well on the average task in a benchmark and poorly on the average task that a worker actually faces.

This issue is especially relevant for generative artificial intelligence. Experiments have found average productivity gains in professional writing and customer support, together with substantial variation across workers and tasks (Noy and Zhang, 2023; Brynjolfsson et al., 2025). In an experiment with management consultants, Dell'Acqua et al. (2026) describe a "jagged technological frontier": access to AI improves performance on some realistic tasks but reduces performance on others. We use *jagged* in this broad sense of unevenness. The term does not require the underlying performance function to jump discontinuously after a small change in a prompt. Reliability can be uneven because some parts of a task space are well covered while others require more interpolation or extrapolation.

In this paper, we ask how a developer or organisation should allocate scarce resources when benchmark performance is easier to observe than deployment exposure. We distinguish five responses. The decision-maker can scale the system broadly, make coverage more uniform, target weak regions, audit the deployment environment, or verify outputs after rollout. These responses address different parts of the same information systems problem. Scaling raises average reliability. Regularity and targeted coverage improve where reliability is delivered. An audit reveals where improvements are needed in a particular workflow. Verification manages the residual tail of errors that remains after deployment.

To see the basic issue, consider a bridge supported by pylons. Suppose that one span is two metres long and another is eight metres long. An engineer who gives one test to each span places weight one-half on each region. A pedestrian crossing the bridge spends one-fifth of the journey on the short span and four-fifths on the long span. If longer spans are also less reliable, the evaluation average understates the pedestrian's exposure to risk. The same distinction arises when a benchmark places equal weight on task categories, but actual use is concentrated in categories where an AI system is weak.

At a statistical level, the distinction resembles covariate shift: the distribution of tasks changes between an evaluation sample and a target environment while conditional loss can remain stable within a sufficiently fine category. The statistical literature develops weighting and validation methods when target exposure can be observed or



Exposure effect: a benchmark tests each span once, but a pedestrian is four times more likely to be on the long span. If longer spans are less reliable, the benchmark understates deployment loss.

Figure 1: Why deployment gives more weight to a long span

estimated (Shimodaira, 2000; Sugiyama et al., 2007), and benchmark work documents consequential distributional shifts in real-world machine learning applications (Koh et al., 2021). Our object is related but different. We take the benchmark–deployment mismatch into an organisational choice problem. Relative deployment exposure determines not only how an average score should be reweighted, but also whether a system should be rolled out, where to direct a scarce reliability budget, whether a workflow audit is valuable, when to verify residual errors, and what a public benchmark should disclose.

Our analysis has three layers. We first study benchmark and deployment weights without imposing a spatial model. Let a fixed set of categories describe the tasks under consideration. Each category has a benchmark weight, a deployment weight, and a local expected loss. Deployment loss differs from benchmark loss by the benchmark-weighted covariance between local loss and relative deployment exposure. This identity is simple, but it organises the economics. A planner who allocates improvement spending using benchmark weights will generally choose a different portfolio from a planner who knows deployment weights. A deployment audit is valuable when it redirects investment or changes the rollout decision. A single public benchmark mean cannot generally support that workflow-specific choice. Category-level reporting can.

We then develop a model of uneven coverage. Knowledge anchors divide a one-dimensional task space into regions. Within a region, the AI system interpolates between known endpoint values. We use Brownian-bridge variance to measure local loss. The assumption yields a clear implication: the average loss within a region is proportional to the length of that region. A user whose task is drawn uniformly from the task line is more likely to arrive in a long region. Under renewal coverage, deployment loss is $(1 + \mathrm{CV}^2)$ times region-uniform benchmark loss, where CV is the coefficient of variation of region lengths. Under Poisson coverage, the multiplier is two.

The uneven-coverage model clarifies the returns to different investments. Increasing scale reduces average region length, but it does not change the relative exposure multiplier when the shape of the region-length distribution is fixed. Increasing regularity reduces the multiplier by compressing the tail of long regions. Adding a targeted anchor in a region of length X has deployment value proportional to X^2. By contrast, under a fixed region-level benchmark, its benchmark value is proportional to X. Both measures can favour large regions, but deployment exposure places a stronger emphasis on the tail. When several anchors can be planned jointly, the optimal placement problem is a discrete allocation problem across original regions rather than a mechanical repetition of one-step midpoint splits.

Finally, we distinguish deployment audits from verification and combine the investment margins in an integrated planning problem. An audit is an ex ante information investment. It estimates the distribution of tasks likely to arise in a particular workflow and, therefore, improves the allocation of a common reliability budget and the rollout decision. Verification is an ex post safeguard. It allows the user or organisation to avoid relying on an output when local loss is too high. These instruments are related but not interchangeable. Better ex ante investment reduces the residual value of verification, but generally does not eliminate it.

Our paper is related to Carnehl and Schneider (2025), who study how available knowledge affects decision quality and how an agent chooses questions to investigate. In contrast, we study the allocation of reliability investment when the evaluation tasks differ from those encountered in deployment. The one-dimensional interpolation model also relates to work on search and learning over correlated landscapes (Callander, 2011; Bardhi and Callander, 2026). We use that structure to provide a microfoundation for uneven reliability, rather than to study search itself.

The distinction between evaluation and deployment also connects to the statistical literature on covariate shift. When the distribution of inputs changes but the conditional relationship between inputs and outcomes remains stable, importance weighting can be used to evaluate or select models for the target distribution (Shimodaira, 2000; Sugiyama et al., 2007). The WILDS benchmark documents that distribution shifts arise across a range of real-world machine-learning applications and can materially reduce out-of-distribution performance (Koh et al., 2021). Our question begins with the recognition that a target distribution matters. We study how scarce information about workflow exposure changes an organisation's decisions: rollout, the allocation of reliability investment, audit incentives, verification, and the information a public benchmark should disclose. Exposure weights are therefore an economic object, not only an input into a statistical correction.

The paper also relates to AI measurement and the organisational use of information systems. Burnell et al. (2023) argue that aggregate evaluation scores can obscure

important variation across capabilities. Guo et al. (2017) study the calibration of modern neural networks, while Lai et al. (2023) survey empirical work on human–AI decision making. Within information systems, the mechanism provides a distributional account of task–technology fit (Goodhue and Thompson, 1995): value depends not only on whether a system performs a task well, but also on how heavily the workflow exposes users to tasks on which performance is weak.

The paper also speaks to the economics of general-purpose technologies and learning by using (Bresnahan and Trajtenberg, 1995; Rosenberg, 1982). An organisation often learns the value of a technology only after adapting it to a workflow. In our setting, adaptation has a distributional dimension: the organisation must learn which tasks matter sufficiently often to justify targeted improvement or verification.

The rest of the paper is organised as follows. Section 2 introduces benchmark and deployment weights and studies targeted investment. Section 3 develops the uneven-coverage model and states the robustness hierarchy. Section 4 compares scale, regularity, and targeted coverage. Section 5 studies deployment audits, verification, and an integrated deployment-planning problem. Section 6 studies public benchmark disclosure and discusses developers, organisations, and empirical implications. Section 7 concludes. Proofs are collected in the Appendix.

2 Benchmark Weights and Deployment Value

We begin with a general environment. The purpose of this section is to separate the economic problem from the spatial model that follows. None of the results in this section requires the assumptions of Brownian motion, Poisson coverage, or a one-dimensional task space that are utilised later.

Table 1 collects the notation used throughout the analysis. To keep the reference compact, related objects are grouped together and notation used only inside proofs is omitted.

2.1 Fixed task categories

There are n fixed task categories, indexed by $i = 1, \dots, n$. A category can be a subject area, a workflow, a class of prompts, or any other partition that is useful for evaluation and deployment. The benchmark assigns weight $b_i > 0$ to category i, with $\sum_i b_i = 1$. Deployment assigns weight $d_i \geq 0$, with $\sum_i d_i = 1$. Let $V_i \geq 0$ denote the task-level loss that would be realised by relying on the AI system after a task from category i arrives, and let

$$v_i = \mathbb{E}[V_i \mid i] \tag{1}$$

Table 1: Notation used in the analysis

Symbol	Definition
Category-level model	
$i = 1, \dots, n$	Task category or region index and number of categories or regions
$b_i, d_i, w_i = d_i / b_i$	Benchmark weight, deployment weight, and relative exposure
V_i; $v_i = \mathbb{E}[V_i \mid i]$; v_i^B, v_i^D	Task-level loss, category mean, and benchmark and deployment conditional losses when they differ
μ_B, μ_D; q; $(x)_+ = \max\{x, 0\}$	Benchmark and deployment mean losses, loss tolerance, and positive-part operator
$\pi_i, \tau_i, \alpha_i(v_i, \tau_i)$	Potential-task share, rules or frictions, and reliance rate
K; a_i; $g_i(a_i)$; $r_i(a_i)$	Reliability budget, targeted spending, improvement, and residual loss
Boldface symbols; $\mathbf{D}$	Corresponding vectors; uncertain deployment-weight vector
$\mathbf{a}^B, \mathbf{a}^D$; η_B, η_D	Benchmark-guided and deployment-aware allocations; associated budget shadow values
$\mathbb{E}, \mathbb{E}_B, \mathbb{E}_D, \mathbb{E}_Z$; $\mathrm{Var}, \mathrm{Cov}, \mathrm{Cov}_B$	Expectation, variance, and covariance; subscripts identify the sampling distribution
Uneven coverage and investment	
L; X, X_i; T	Task-line length, region length, and relative task location
λ; CV; X^*; $\bar{v}(X)$	Coverage density, coefficient of variation, deployment-encountered region length, and average loss in a region
α; $\Delta_D(X, \alpha), \Delta_B(X_i, \alpha)$	Anchor split fraction and deployment and benchmark gains
$C_S(\lambda), C_R(\mathrm{CV})$	Costs of scale and regularity
M; $m_i \in \mathbb{N}_0$; $\Delta_i(m_i)$	Total and region-level planned anchors; marginal gain from another anchor
Audits, verification, and the integrated planner	
Z, Z'; $V(Z)$; C_A; K'	Audit signals, audit value, audit cost, and remaining budget
δ; a_H	Exposure imbalance and high-exposure spending in the two-category example
s; $h(s)$; $C(s)$; $\theta(s)$	Scale spending, additive improvement, cost, and proportional improvement
$\bar{c}$; c_i	Blanket and category-specific verification costs
ρ; $e(\rho)$; ω_i	Regularity spending, improvement, and category loading
$A(Z)$; $\tilde{d}_i(Z)$	Resource cost of an audit and posterior expected exposure
L_i^0; $L_i(s, \rho, a_i)$	Baseline and residual task-level losses
$\ell_i(s, \rho, a_i)$; y_i; u_i; $W(Z)$	Mean residual loss, verification choice, category payoff, and integrated value
ψ_i; η	Share benefiting from a downward loss shift and common-budget shadow value
Reporting and extensions	
j; $C \in \mathcal{P}$; b_C, d_C	Primitive task type, reporting cell and partition, and cell weights
$\bar{v}_C$; $\hat{\mu}_D(\mathcal{P})$; σ_C^2; κ	Reported cell mean, partition-based workflow-loss estimate, estimation variance, and disclosure-cost parameter
$G_i(a_1, \dots, a_n)$	General loss technology allowing cross-category spillovers
$S = \{i : b_i > 0\}$; X'; ζ	Benchmark-supported categories, independent copy of X, and proportional-loss constant in the Appendix

denote the category-level expected loss. Most ex ante investment and rollout calculations use the mean v_i. Verification after rollout uses the distribution of V_i within the category, because it screens realised tasks rather than category averages. The category-level model is recovered as the degenerate case in which $V_i = v_i$ almost surely.

The category-level notation assumes that, conditional on category i, the expected loss in the benchmark and deployment environments is the same. This is a granularity requirement, not a substantive claim that coarse categories are always sufficient. If the within-category task mix differs between the benchmark and deployment environments, the categories should be refined until the conditional loss stabilises. Equivalently, if benchmark and deployment conditional losses are v_i^B and v_i^D, then the total wedge decomposes as

$$\sum_{i=1}^{n} d_i v_i^D - \sum_{i=1}^{n} b_i v_i^B = \underbrace{\sum_{i=1}^{n} d_i (v_i^D - v_i^B)}_{\text{within-category shift}} + \underbrace{\mathrm{Cov}_B(w_i, v_i^B)}_{\text{exposure wedge}}, \tag{2}$$

where $w_i = d_i / b_i$. The analysis below focuses on the case in which the within-category shift is zero because categories are defined finely enough for $v_i^B = v_i^D = v_i$.

The benchmark and deployment averages are, respectively,

$$\mu_B = \sum_{i=1}^{n} b_i v_i \qquad \text{and} \qquad \mu_D = \sum_{i=1}^{n} d_i v_i. \tag{3}$$

The first average is the object that an evaluator reports. The second is the object that governs users' experience in a particular deployment environment.

For each category, define relative deployment exposure as $w_i = d_i / b_i$. This ratio is greater than one when a category appears more frequently in deployment than in the benchmark. The following proposition gives the paper's basic accounting identity.

Proposition 1 (Exposure wedge)**.** *The difference between deployment loss and benchmark loss is*

$$\mu_D - \mu_B = \mathrm{Cov}_B(w_i, v_i). \tag{4}$$

Hence, deployment loss exceeds benchmark loss if and only if the benchmark-weighted covariance between local loss and relative deployment exposure is positive.[1]

The interpretation of Proposition 1 is straightforward. A benchmark is optimistic when it underweights the tasks on which the system is weak. This statement does not require tasks to lie on a line, nor does it require any particular source of unevenness. The bridge example in Figure 1 is one possible microfoundation. An enterprise coding assistant, for instance, can face the same problem if public evaluations place relatively little weight on the legacy languages that appear frequently in the enterprise's code base.

Remark 1 (Missing support)**.** *The assumption $b_i > 0$ keeps the relative-exposure ratio well defined. If a category has $d_i > 0$ but $b_i = 0$, the benchmark has a coverage omission rather than a weighting error. Its deployment relevance cannot be recovered by reweighting observed benchmark categories.*

The remark distinguishes two measurement problems. A weighting error can be corrected when the evaluator observes category-level performance and learns deployment exposure. A coverage omission requires new evaluation tasks.

2.2 Why deployment weights are scarce information

Benchmark weights are usually visible by construction. An evaluator knows how many coding tasks, legal questions, or customer service interactions are included in an evaluation suite. Deployment weights are different. They depend on the workflow in

[1]The covariance wording is deliberate. If relative exposure or local loss is constant, a correlation coefficient can be undefined even though the covariance identity remains valid.

which the model is used, the users who adopt the model, and the tasks for which those users choose to rely on it. A developer serving many organisations may observe only a noisy average of these environments. An organisation preparing a rollout may not yet possess usage logs for a technology that has not been deployed.

This distinction does not imply that deployment weights are immutable. When a model becomes more reliable in one category, users can expand its use in that category. When an organisation introduces a verification requirement, some tasks become less attractive to automate. The main model takes deployment weights as fixed during a single investment and rollout decision. This is a short-run approximation that isolates the initial measurement problem.

A minimal endogenous-use extension gives the same exposure logic. Let π_i denote the arrival share of potential tasks before users decide whether to rely on the AI system. Let $\alpha_i(v_i, \tau_i) \in [0, 1]$ denote the reliance rate in category i, where τ_i summarises review rules, organisational permissions, or user frictions. Realised exposure is then

$$d_i(\mathbf{v}, \boldsymbol{\tau}) = \frac{\pi_i \alpha_i(v_i, \tau_i)}{\sum_{j=1}^{n} \pi_j \alpha_j(v_j, \tau_j)}. \tag{5}$$

For any realised vector of reliance rates, the wedge remains

$$\mu_D(\mathbf{v}, \boldsymbol{\tau}) - \mu_B = \mathrm{Cov}_B \left(\frac{d_i(\mathbf{v}, \boldsymbol{\tau})}{b_i}, v_i \right). \tag{6}$$

Endogenous use can shrink or reverse the wedge if users disproportionately avoid weak categories. It does not mechanically eliminate the wedge, because potential task arrivals, organisational necessity, and verification frictions can still leave weak categories overrepresented in realised reliance. In this interpretation, an audit estimates the post-rule exposure weights relevant for the immediate decision rather than immutable long-run demand.

The categories can also incorporate differences in stakes. A rare task with a costly failure can matter as much as a frequent task with a modest failure. For simplicity, we place the expected consequence of a failure inside the local-loss term v_i. The deployment weight d_i then measures frequency, while v_i measures the expected loss conditional on encountering the category. Equivalently, a researcher interested only in error rates can interpret v_i as a welfare-weighted error rate.

Finally, benchmark weights need not be uniform. Some evaluations are deliberately tilted towards safety-critical tasks, while others sample from observed product use. Proposition 1 accommodates both designs. The relevant question is not whether a benchmark is balanced in an abstract sense. It is whether its weights are appropriate for the deployment decision under consideration.

2.3 Rollout decisions

We next translate the exposure wedge into a rollout decision. Suppose that relying on the AI system rather than the outside option creates a gross benefit $q > 0$ and an expected loss v_i in category i. The deployment payoff from rollout is therefore $q - \mu_D$. We refer to q as the *loss tolerance*. A higher value of q captures a lower-stakes task, an easier manual correction, or a larger productivity benefit.

A deployment-aware planner, therefore, rolls out the AI system if and only if $q \geq \mu_D$. A planner who uses the benchmark average instead rolls out if and only if $q \geq \mu_B$. If $\mu_D > \mu_B$, benchmark-based over-rollout occurs when

$$\mu_B \leq q < \mu_D. \tag{7}$$

If $\mu_B > \mu_D$, benchmark-based under-rollout occurs when

$$\mu_D \leq q < \mu_B. \tag{8}$$

This is a selection effect. The issue is not that an organisation irrationally ignores an obvious statistical fact. Rather, the organisation may observe benchmark loss more readily than deployment loss. If the system sits near the rollout threshold, a modest difference between the two distributions can change the decision.

2.4 Targeted improvement

The exposure wedge also changes the preferred allocation of improvement spending. Suppose that the planner has a fixed adaptation budget $K > 0$. Spending $a_i \geq 0$ in category i lowers loss by $g_i(a_i)$, where g_i is increasing, concave, and differentiable over the relevant range. The feasible set is $\sum_i a_i \leq K$. This formulation can capture domain-specific fine-tuning, retrieval support, evaluation-driven repair, or workflow adaptation. To avoid negative residual losses, one can interpret g_i as effective improvement before the zero-loss floor binds, or replace residual loss below by $(v_i - g_i(a_i))_+$. The marginal conditions below apply directly on ranges where the floor is not binding; a binding floor is a corner case in the same Kuhn–Tucker system.

The deployment-aware planner chooses spending to maximise the reduction in deployment loss:

$$\max_{\{a_i\}_{i=1}^n} \sum_{i=1}^{n} d_i g_i(a_i) \qquad \text{subject to} \qquad \sum_{i=1}^{n} a_i \leq K. \tag{9}$$

By contrast, a planner guided by benchmark weights replaces d_i with b_i. Thus, the same technological opportunities can generate different investment portfolios because

the two planners attach different values to local improvement.

Because the objective in (9) is concave, the Kuhn–Tucker conditions characterise the optimum. Let $\mathbf{a}^D (\equiv \{a_i^D\}_{i=1}^n)$ denote a deployment-aware allocation. There exists a multiplier $\eta_D \geq 0$ such that

$$d_i g_i'(a_i^D) \leq \eta_D, \tag{10}$$

$$a_i^D \left(\eta_D - d_i g_i'(a_i^D)\right) = 0, \tag{11}$$

$$\eta_D \left(K - \sum_{i=1}^{n} a_i^D\right) = 0. \tag{12}$$

Whenever $a_i^D > 0$ and the loss floor is not binding, $d_i g_i'(a_i^D) = \eta_D$. Benchmark-guided spending satisfies the analogous conditions with b_i and multiplier η_B in place of d_i and η_D. If both solutions are interior and the budget binds, the conditions reduce to

$$d_i g_i'(a_i^D) = \eta_D \qquad \text{and} \qquad b_i g_i'(a_i^B) = \eta_B \tag{13}$$

for all categories. Relative deployment exposure d_i / b_i therefore determines which categories receive more attention once the planner has learned the deployment distribution.

The interpretation is a marginal-value effect. An additional dollar spent in category i matters in proportion to the number of tasks exposed to that improvement. A benchmark-guided planner and a deployment-aware planner face the same engineering frontier, but they value movement along that frontier differently. The distinction is particularly relevant for an organisation adapting a general-purpose model to a narrow workflow.

To fix ideas, suppose there are two categories and the benchmark assigns equal weight to them. If deployment places three-quarters of the weight on the first category and one-quarter on the second, an audit revealing those weights raises the marginal value of improvement in the first category relative to the second. Even when the benchmark correctly identifies the weaker category, it need not attach the correct intensity to the repair.

2.5 A general planner problem

The rollout and investment decisions can be combined. Before deployment, a planner chooses an adaptation allocation $\mathbf{a} = (a_1, \ldots, a_n)$ subject to the budget constraint. After adaptation, residual loss in category i is

$$r_i(a_i) = \big(v_i - g_i(a_i)\big)_+. \tag{14}$$

If the deployment weights are known, the planner chooses **a** and rollout to maximise

$$\left(q - \sum_{i=1}^{n} d_i r_i(a_i) \right)_+ . \tag{15}$$

The positive-part operator captures the option not to deploy. A benchmark-guided planner uses b_i in place of d_i. Thus, the measurement problem affects both the intensive margin of investment and the extensive margin of rollout.

This formulation clarifies why the allocation question is not exhausted by the covariance identity. The identity identifies the direction of benchmark error for a given model. The planner problem asks how that error changes a decision made before deployment. A benchmark can lead a planner to deploy too early, to invest in the wrong categories, or to do both. The following sections develop the model in which these choices can be compared explicitly.

3 A Model of Uneven Coverage

The preceding analysis treated local losses as primitives. We now provide a microfoundation for a positive association between local loss and deployment exposure. The model isolates a concrete and common source of unevenness, irregular coverage of a task space, and turns it into sharp predictions about where deployment loss concentrates.

3.1 Interpolation within a coverage region

Consider a long one-dimensional task line. Effective knowledge anchors divide the line into regions. An anchor is a location at which the model has sufficient support to make interpolation loss negligible. This support can reflect training data, retrieval, representation quality, or a combination of these objects. An anchor represents effective support, not the memorisation of an isolated prompt.

Within a region of length X, let $T \in [0, 1]$ denote the relative location of a task. We use a Brownian landscape to describe the unknown truth between the two anchors. Conditional on the endpoint values, the posterior variance at the task is the Brownian-bridge variance. The Brownian variance rate is normalised to one; a different variance rate would multiply all losses by a constant and would not change the exposure ratios below.

Assumption 1 (Linear interpolation loss)**.** *Within a coverage region of length X, the local expected loss at relative location T is*

$$v(X, T) = XT(1 - T). \tag{16}$$

Here, Assumption 1 captures sparse interpolation risk when loss is measured by squared error or posterior variance. Loss is zero at an idealised anchor and highest at the midpoint of a region. Nearby tasks have correlated truth values, but uneven anchor spacing generates uneven reliability. Brownian-bridge posterior variance provides a microfoundation for the assumption (Karatzas and Shreve, 1991). It captures a clear and important driver of unevenness: reliability degrades with distance from well-supported tasks, so sparsely covered regions are the least reliable.

If a task is drawn uniformly from a given region, T is uniformly distributed on $[0, 1]$. The average loss in a region of length X is therefore

$$\mathbb{E}[v(X, T) \mid X] = X\mathbb{E}[T(1 - T)] = \frac{X}{6}. \tag{17}$$

Thus, longer regions are more error-prone on average. This linear relationship is the only role played by the Brownian-bridge structure in the main results below.

3.2 A finite task line

Before turning to renewal coverage, it is useful to state the finite version of the argument. Suppose that anchors divide a task line of total length L into regions with lengths $X_1, \ldots, X_n$, where $\sum_i X_i = L$. A region-uniform benchmark gives weight $1/n$ to each region. Uniform deployment along the task line gives weight X_i/L to region i.

Using (17), benchmark and deployment loss are

$$\mu_B = \frac{1}{6n} \sum_{i=1}^{n} X_i \qquad \text{and} \qquad \mu_D = \frac{1}{6L} \sum_{i=1}^{n} X_i^2. \tag{18}$$

The first expression depends only on average region length. The second depends on the second moment of region length. In the bridge example, the benchmark average region length is $(2 + 8)/2 = 5$, while the deployment average is $(2^2 + 8^2)/(2 + 8) = 6.8$. The difference arises because the longer region carries more deployment mass.

The finite calculation also shows that the uneven-coverage model is an instance of Proposition 1. Region i has benchmark weight $1/n$, deployment weight X_i/L, and average loss $X_i/6$. Relative deployment exposure and conditional loss therefore rise together with region length.

3.3 Region-uniform benchmarks and deployment exposure

We now distinguish two ways to sample regions. A region-uniform benchmark draws one representative region and then draws a task uniformly within that region. Deployment draws a task uniformly from the task line. The second procedure is more likely to

select a long region because a long region occupies more of the line.

Assumption 2 (Renewal coverage)**.** *Coverage-region lengths are independent draws from a distribution with finite first and second moments. Let X denote a representative region length, with*

$$\mathbb{E}[X] = \frac{1}{\lambda} \qquad \textit{and} \qquad \mathrm{CV}^2 = \frac{\mathrm{Var}(X)}{\mathbb{E}[X]^2}. \tag{19}$$

The parameter λ measures average coverage density. The coefficient of variation CV measures irregularity. At one boundary, $\mathrm{CV} = 0$ corresponds to equally spaced anchors. The Poisson case considered below has $\mathrm{CV} = 1$. Assumption 2 is for tractability. It provides a transparent benchmark in which mean coverage and the dispersion of coverage can be varied separately.

Let X^* denote the region containing a uniformly drawn deployment task. The distribution of X^* is length-biased: a region of length x is selected in proportion to x.

Lemma 1 (Renewal length bias)**.** *In the stationary renewal interpretation, or in the large-domain limit of a finite renewal task line, the average region length encountered in deployment is*

$$\mathbb{E}[X^*] = \frac{\mathbb{E}[X^2]}{\mathbb{E}[X]} = \mathbb{E}[X](1 + \mathrm{CV}^2). \tag{20}$$

Lemma 1 is the classical *inspection paradox* for renewal intervals (Ross, 1996). The paradox is familiar from waiting times. A passenger arriving at a bus stop at a random time is more likely to arrive during a long interval between buses than during a short interval. Consequently, the interval inspected by the passenger is longer on average than the interval obtained by selecting one bus interval uniformly from a timetable. There is no contradiction. Long intervals simply occupy more time and are therefore more likely to contain a random arrival.

The same logic applies to the bridge example in Figure 1. A region-uniform benchmark selects the two-metre span and the eight-metre span with equal probability. A pedestrian crossing the bridge is four times more likely to be located on the eight-metre span because it occupies four times as much of the route. More generally, a region of length x receives deployment weight in proportion to x. This is why the average region encountered in deployment is $\mathbb{E}[X^2]/\mathbb{E}[X]$ rather than $\mathbb{E}[X]$.

The inspection paradox has a precise role in our paper. By itself, it is a statement about exposure, not error. It says that users are disproportionately exposed to long regions. Assumption 1 supplies the second step: long regions also have higher average interpolation loss. Combining these two steps implies that deployment gives greater weight to the parts of the task space where the AI system is less reliable. Proposition 2 below quantifies the resulting benchmark-deployment wedge.

The connection to public benchmarks requires some care. A public benchmark does not literally observe renewal regions or draw one question from every latent

coverage gap. However, many public benchmarks are designed to represent a range of capabilities, domains, or task types rather than the frequency with which those tasks arise in a particular workflow. Such a benchmark is closer to region-uniform sampling than to deployment-weighted sampling. The inspection paradox identifies one reason this difference can matter: large, poorly covered regions can receive more user exposure than their representation in a benchmark would suggest. The exact multiplier $1 + \mathrm{CV}^2$ belongs to the one-dimensional model. The broader lesson is the general exposure condition in Proposition 1: a public benchmark is optimistic for a given use case when it underweights categories in which local loss is high.

Proposition 2 (Exposure multiplier)**.** *Under Assumptions 1 and 2, region-uniform benchmark loss and deployment loss are*

$$\mu_B = \frac{1}{6\lambda} \qquad \textit{and} \qquad \mu_D = \frac{1 + \mathrm{CV}^2}{6\lambda}. \tag{21}$$

Consequently,

$$\mu_D = (1 + \mathrm{CV}^2)\mu_B. \tag{22}$$

The interpretation of Proposition 2 follows from two forces. The first is an exposure effect: deployment selects long regions more often than a region-uniform benchmark. The second is an interpolation effect: the average local loss is higher in a long region. The multiplier combines these forces. At the regular boundary $\mathrm{CV} = 0$, benchmark and deployment loss coincide. As dispersion rises, the deployment distribution places increasing weight on the least-covered parts of the task space.

The exact multiplier in (22) uses the linear average-loss implication of Assumption 1. The direction of the wedge is more general. If average loss in a region of length X is $\bar{v}(X)$, then region-uniform benchmark loss is $\mathbb{E}[\bar{v}(X)]$, deployment loss is $\mathbb{E}[X\bar{v}(X)]/\mathbb{E}[X]$, and the difference is

$$\frac{\mathrm{Cov}(X, \bar{v}(X))}{\mathbb{E}[X]}. \tag{23}$$

Thus, size-biased exposure raises deployment loss whenever average local loss has positive covariance with region length.

The variable X should not be read as requiring an organisation to observe literal latent anchor distances. In applications, the operational analogue is any observable measure that predicts sparse support: nearest-neighbour distance in an embedding space, retrieval similarity, representation density, frequency in historical logs, disagreement across models or reviewers, or empirical loss in a workflow pilot. These proxies need not recover the Brownian geometry. They are sufficient for the economic argument when they identify categories for which exposure and loss are positively associated. The exact multiplier belongs to the clean one-dimensional model; the sign and allocation logic belong to the broader exposure condition.

3.4 Poisson coverage

The factor of two arises as a special case. Suppose that knowledge anchors are generated by a homogeneous Poisson process with intensity λ. Consecutive region lengths are then exponentially distributed with mean $1/\lambda$ and coefficient of variation one.

Corollary 1 (Poisson factor of two)**.** *Under Poisson coverage and Assumption 1,*

$$\mu_B = \frac{1}{6\lambda}, \qquad \mu_D = \frac{1}{3\lambda}, \qquad \textit{and} \qquad \mu_D = 2\mu_B. \tag{24}$$

Corollary 1 is useful because it gives a memorable quantitative illustration. It is not the general economic result. Poisson coverage is sufficient for a factor-of-two gap-length ratio. Translating that length ratio into a factor-of-two loss ratio additionally requires average loss to be proportional to region length, as in Assumption 1.

3.5 Robustness hierarchy

The model has a deliberate hierarchy. The benchmark-deployment wedge is a general exposure-weight result and does not depend on Brownian motion, Poisson coverage, or a one-dimensional task space. The renewal model supplies a tractable account of size-biased exposure. Brownian-bridge interpolation maps larger regions into proportionally larger average losses. Poisson coverage then yields the factor-of-two special case. Table 2 summarises which assumptions support each result.

The last row is intentionally qualitative. In more than one dimension, a randomly located task is disproportionately likely to fall in a large, sparse cell. However, local loss depends on cell geometry and on the interpolation rule. The exact multiplier is therefore model-specific. We return to this distinction in the Appendix.

3.6 A stationary interpretation

The task line can be interpreted as a large finite interval, with a task drawn uniformly before taking a large-domain limit. Equivalently, one can place a representative deployment task at the origin and study the nearest anchors to its left and right. In this stationary local interpretation, the origin is only a coordinate normalisation. It is not an anchor at which the truth is known.

This point is important for the Brownian formulation. A two-sided Brownian motion is often normalised to equal zero at the origin. That mathematical normalisation should not be confused with an observation available to the model. The long-interval formulation keeps the economic object clear: uncertainty is determined by the two neighbouring anchors around a representative task.

Table 2: Robustness hierarchy

Result	Requires Brownian bridge?	Requires Poisson?	Requires one dimension?
Density-ratio wedge: $\mathbb{E}_D[v] - \mathbb{E}_B[v] = \mathrm{Cov}_B(w, v)$	No	No	No
Finite exposure wedge: $\sum_i d_i v_i - \sum_i b_i v_i$	No	No	No
Renewal length bias: $\mathbb{E}[X^*]/\mathbb{E}[X] = 1 + \mathrm{CV}^2$	No	No	Yes, for interval gaps
Finite $(1 + \mathrm{CV}^2)$ loss ratio	No, but requires average loss proportional to region length; the bridge gives $X/6$	No	Yes, as stated
Poisson factor-of-two length ratio	No	Yes	Yes, as stated
Poisson factor-of-two loss ratio	No, but requires average loss proportional to region length	Yes	Yes, as stated
Multidimensional size bias towards large sparse cells	No	No	No, but exact constants change

4 Investing in Deployment Reliability

We now use the uneven-coverage model to compare three ex ante investments: scale, regularity, and targeted coverage. The purpose is not to assign a universal engineering interpretation to any one parameter. Rather, the model identifies three ways to improve reliability, each with different consequences for benchmark performance and deployment value.

4.1 Scale and regularity

Recall from Proposition 2 that deployment loss is

$$\mu_D(\lambda, \mathrm{CV}) = \frac{1 + \mathrm{CV}^2}{6\lambda}. \tag{25}$$

This expression separates two forces. Scale raises λ and shortens the average region. Regularity lowers CV and compresses the tail of long regions.

The marginal reduction in deployment loss from increasing scale is

$$-\frac{\partial \mu_D}{\partial \lambda} = \frac{1 + \mathrm{CV}^2}{6\lambda^2}. \tag{26}$$

The marginal reduction from lowering irregularity is

$$\frac{\partial \mu_D}{\partial \mathrm{CV}} = \frac{\mathrm{CV}}{3\lambda}. \tag{27}$$

The second expression is written as the loss reduction from a one-unit decline in CV. Scale improves the level of reliability. Regularity reduces the exposure penalty generated by the tail.

Holding CV fixed, increasing λ lowers benchmark and deployment loss proportionally but leaves the relative exposure multiplier $1 + \mathrm{CV}^2$ unchanged. Holding λ fixed, lowering CV reduces deployment loss and the multiplier. Dividing (27) by (26) gives the ratio of the marginal deployment-loss reduction from a one-unit decline in CV to the marginal deployment-loss reduction from a one-unit increase in λ:

$$\frac{2\,\mathrm{CV}\,\lambda}{1 + \mathrm{CV}^2}. \tag{28}$$

For $\mathrm{CV} > 0$, this ratio rises with λ. The comparison describes a level-versus-shape trade-off. Early in development, broad scaling can be valuable because average regions are long. As scale increases, its marginal effect falls at a rate λ^{-2}. The marginal value of regularity falls more slowly, at a rate λ^{-1}. The ratio in (28) is expressed in the model's units; the cost-adjusted comparison is the portfolio condition below. Put differently, after broad coverage has improved substantially, the remaining long regions deserve more attention.

This comparison provides a disciplined way to discuss curation, active failure discovery, and retrieval design. These interventions can make coverage more regular without simply increasing the volume of support. In the model, moving from exponential spacing, where $\mathrm{CV} = 1$, to deterministic spacing, where $\mathrm{CV} = 0$, halves deployment loss at a fixed mean spacing. We do not interpret this calculation as a universal conversion rate between synthetic and human-generated data. Mapping engineering choices into λ and CV is an empirical question.

4.2 Targeted coverage

Regularity is an aggregate property. We next consider a local intervention. Suppose that a finite task line of total length L is divided into regions with lengths $X_1, \ldots, X_n$. Under uniform deployment exposure along the line, Proposition 2 has the finite counterpart

$$\mu_D = \frac{1}{6L} \sum_{i=1}^{n} X_i^2. \tag{29}$$

The square in this expression captures the combined exposure and interpolation effects. A long region is encountered more often and has a higher conditional loss.

Suppose that one additional anchor is placed inside a region of length X. Let $\alpha \in (0,1)$ be the fraction of the original region lying to the left of the new anchor. The new regions have lengths αX and $(1-\alpha)X$.

Proposition 3 (Targeted repair)**.** *Adding an anchor that splits a region of length X at α lowers deployment loss by*

$$\Delta_D(X,\alpha) = \frac{\alpha(1-\alpha)X^2}{3L}. \tag{30}$$

The gain is largest at the midpoint, where $\alpha = 1/2$.

Proposition 3 describes a tail effect. Closing a large region has disproportionate deployment value because it improves many tasks and because those tasks were relatively error-prone before the intervention. The midpoint result is also intuitive: a new anchor is most valuable when it balances the two remaining interpolation problems.

4.3 Fixed benchmark correction

The comparison with a benchmark requires care. A benchmark should not be redefined each time the model gains an anchor. We therefore fix the original regions as benchmark strata. The benchmark draws one stratum uniformly, then draws one task uniformly within that stratum. If an anchor is added, the benchmark still samples from the same original stratum.

Proposition 4 (Benchmark and deployment targeting)**.** *Under a fixed region-level benchmark, adding an anchor that splits the original region i at α lowers benchmark loss by*

$$\Delta_B(X_i,\alpha) = \frac{\alpha(1-\alpha)X_i}{3n}. \tag{31}$$

Its deployment gain is given by (30)*. Hence, the benchmark value is proportional to X_i, while the deployment value is proportional to X_i^2.*

The interpretation of Proposition 4 is not that benchmarks attach equal value to every improvement. A fixed benchmark can correctly indicate that a larger weak region deserves attention. The sharper point is that deployment places a stronger weight on the tail. A planner using benchmark data alone is likely to understate how aggressively resources should be directed towards the largest high-exposure regions.

4.4 An investment portfolio

The three ex ante margins can now be compared. Begin with scale and regularity. Let $C_S(\lambda)$ denote the differentiable cost of achieving coverage density λ, with $C_S'(\lambda) > 0$.

Let $C_R(\mathrm{CV})$ denote the differentiable cost of achieving irregularity CV, with $C_R'(\mathrm{CV}) < 0$: making coverage more regular is costly. A planner with a reliability budget K solves

$$\min_{\lambda,\mathrm{CV}} \frac{1+\mathrm{CV}^2}{6\lambda} \qquad \text{subject to} \qquad C_S(\lambda) + C_R(\mathrm{CV}) \le K. \tag{32}$$

The constraint describes a resource-allocation problem. Increasing scale uses a budget. Reducing irregularity also uses the budget. At an interior local optimum with a binding budget, the planner equalises the loss reduction generated by the last dollar spent on each margin.

At an interior local optimum of (32) with a binding budget constraint, the planner equalises the loss reduction generated by the last dollar spent on each margin:

$$\frac{(1+\mathrm{CV}^2)/(6\lambda^2)}{C_S'(\lambda)} = \frac{\mathrm{CV}/(3\lambda)}{-C_R'(\mathrm{CV})}. \tag{33}$$

The left-hand side is the deployment-loss reduction from the last dollar spent on scale. The right-hand side is the deployment-loss reduction from the last dollar spent on regularity. The cost-adjusted return to regularity relative to scale is

$$\frac{\left[\mathrm{CV}/(3\lambda)\right]/\left[-C_R'(\mathrm{CV})\right]}{\left[(1+\mathrm{CV}^2)/(6\lambda^2)\right]/C_S'(\lambda)} = \frac{2\lambda\,\mathrm{CV}\,C_S'(\lambda)}{(1+\mathrm{CV}^2)\left[-C_R'(\mathrm{CV})\right]}. \tag{34}$$

Holding CV and the marginal costs fixed, this ratio rises with λ: in model units, the direct marginal return to scale falls at rate λ^{-2}, while the direct marginal return to regularity falls at rate λ^{-1}. The portfolio comparative static is weaker. Along an engineering path, $C_S'(\lambda)$, $-C_R'(\mathrm{CV})$, and CV can move as coverage improves, so (33) rather than the unadjusted rate comparison determines the preferred mix. The safe conclusion is that broad scaling remains valuable when many regions are long, while the case for regularity strengthens only under marginal-cost and path conditions that do not offset the physical decline in scale returns.

Targeted repair adds a third margin. Suppose that the planner can place a limited number of additional anchors after observing the original regions. Proposition 3 implies a one-anchor rule and a separate multi-anchor allocation problem.

Corollary 2 (Targeted-anchor placement)**.** *If the planner can add one anchor, it should split the longest original region at its midpoint. If the planner can plan $M \ge 1$ anchors jointly, and if $m_i \in \mathbb{N}_0$ anchors are assigned to the original region i, then those m_i anchors should divide that region into $m_i + 1$ equal subregions. The optimal integer allocation solves*

$$\min_{m_1,\ldots,m_n \in \mathbb{N}_0} \sum_{i=1}^{n} \frac{X_i^2}{m_i+1} \qquad \textit{subject to} \qquad \sum_{i=1}^{n} m_i = M. \tag{35}$$

Equivalently, because marginal gains are decreasing within each original region, the M planned anchors can be assigned by repeatedly choosing a region i with the largest current planned marginal gain

$$\Delta_i(m_i) = \frac{X_i^2}{6L(m_i+1)(m_i+2)}, \tag{36}$$

and then placing the assigned anchors evenly within each of the original regions. If anchors must be placed irreversibly, one at a time, and cannot later be repositioned, then splitting the longest current region at its midpoint is a one-step optimal rule but not generally a globally optimal multi-anchor plan.

The one-anchor priority rule follows from the quadratic term in (30). A midpoint split of region X lowers deployment loss by $X^2/(12L)$. Hence, an anchor placed in a region twice as long creates four times the one-step deployment gain. The multi-anchor formula shows why joint planning and irreversible sequential placement are different problems. In a single region of length X, two jointly planned anchors divide the interval into thirds and leave squared-length sum $X^2/3$. A midpoint split followed by an irreversible split of one half leaves squared-length sum $3X^2/8$, which is larger.

The bridge example makes the comparison with a fixed benchmark concrete. A midpoint anchor in the two-metre region lowers its within-stratum average loss by $2/12$. A midpoint anchor in the eight-metre region lowers its within-stratum average loss by $8/12$. Thus, the fixed benchmark values the long-region repair four times as much. Deployment places an additional weight on exposure: the corresponding deployment gains are proportional to 2^2 and 8^2. It values the long-region repair sixteen times as much. The benchmark points in the correct direction but understates the intensity of the preferred response.

A planner with a fixed reliability budget, therefore, chooses among three types of improvement. Scale is appropriate when loss is broadly distributed. Regularity is appropriate when the region-length distribution has a persistent tail. Targeted repair is appropriate when the planner can identify particular high-exposure weak regions. The final condition matters: without information about deployment exposure, a planner can improve the system while still choosing the wrong portfolio.

Table 3 summarises the division of labour among the instruments. Scale has the least demanding informational requirement because it broadly increases reliability. Targeted coverage has the most demanding ex ante requirement because the planner must know where repairs are needed. Verification requires local information later, after a task arrives.

This observation links the uneven-coverage model back to the exposure-weighted adaptation conditions in Section 2. Benchmark weights determine the investment portfolio when they are the planner's best available guide. Deployment weights determine the portfolio that maximises realised value. The next section studies the

Table 3: Reliability investments in the uneven-coverage model

Instrument	Primary effect	Information needed	Timing
Scale	Shortens regions broadly by raising λ	Broad measures of capability and cost	Before deployment
Regularity	Compresses the tail by lowering CV	Distribution of weak regions	Before deployment
Targeted coverage	Splits particular high-loss regions	Local exposure and loss	Before deployment
Deployment audit	Estimates the deployment distribution	Workflow sample or pilot	Before investment and rollout
Verification	Screens residual high-loss tasks	Task-level signal or review	After a task arrives

value of learning the difference.

5 Deployment Audits and Verification

We now introduce information acquisition. A deployment audit and output verification both respond to uncertainty, but they operate at different stages. An audit is conducted before investment and rollout. Verification is applied after rollout to tasks that remain error-prone.

5.1 The value of a deployment audit

Return to the fixed categories in Section 2. The benchmark weights and category-level losses are observed, but the deployment distribution is initially uncertain. Let $\mathbf{D} = (d_1, \ldots, d_n)$ denote the deployment weights and let Z denote a signal produced by an audit. The signal can be based on workflow logs, a pilot programme, historical tasks, or a structured sample of intended use.

For an adaptation allocation $\mathbf{a} = (a_1, \ldots, a_n)$ satisfying $\sum_i a_i \leq K$, residual deployment loss is

$$\mu_D(\mathbf{a}) = \sum_{i=1}^{n} d_i r_i(a_i). \tag{37}$$

After observing an audit signal Z, the planner chooses the adaptation allocation and rolls out if the posterior expected payoff is non-negative. The value before audit cost is therefore

$$V(Z) = \mathbb{E}_Z \left[\max_{\{a_i\}: \sum_i a_i \leq K} (q - \mathbb{E}[\mu_D(\mathbf{a}) \mid Z])_+ \right]. \tag{38}$$

The positive-part operator captures the rollout decision. The maximisation captures the investment decision. An audit can therefore create value through two channels.

Proposition 5 (Deployment information)**.** *Suppose that audit signal Z' is more informative than audit signal Z in the sense of Blackwell (1953). Then*

$$V(Z') \geq V(Z). \tag{39}$$

The inequality is strict when the additional information permits a decision rule that yields strictly greater expected payoff than every decision rule measurable with respect to Z. A change in the selected allocation or rollout decision is therefore sufficient for strictness only when it is payoff-relevant, rather than merely a tie-breaking change. An audit with cost C_A is worthwhile if $V(Z) - V(\varnothing) > C_A$.

The interpretation of Proposition 5 follows from two effects. The *allocation effect* arises when an audit redirects adaptation towards categories with high relative exposure to deployment. The *selection effect* arises when an audit changes whether rollout is worthwhile. The allocation effect matters even when rollout is certain. The selection effect is strongest near the rollout threshold.

The audit cost can also be treated as part of the same resource constraint as engineering. Write $V(Z; K')$ for the value of signal Z when the remaining reliability budget is K'. If an audit with resource cost C_A consumes the common budget rather than being financed separately, the audit is worthwhile when

$$V(Z; K - C_A) > V(\varnothing; K). \tag{40}$$

This formulation makes information acquisition an explore–exploit decision: a broader or more precise audit can improve allocation and selection, but it leaves fewer resources for scale, regularity, targeted repair, or verification capacity.

To fix ideas, consider two categories with equal benchmark weights. Suppose that the actual deployment assigns weights $1/2 + \delta$ and $1/2 - \delta$ to the two categories, with $\delta \in [0, 1/2]$, but the planner does not know in advance which category receives the larger weight. Let improvement in category i be $\sqrt{a_i}$, suppose the loss floor does not bind over the relevant allocations, and let the adaptation budget be K. Before an audit, symmetry implies $a_1 = a_2 = K/2$. After a perfectly informative audit, the high-exposure category receives

$$a_H = \frac{K(1/2 + \delta)^2}{(1/2 + \delta)^2 + (1/2 - \delta)^2}. \tag{41}$$

The audit shifts spending towards the category that users are more likely to encounter. As δ rises, the preferred portfolio becomes more uneven and the allocation value of the audit increases.

Proposition 6 (Allocation value of an audit)**.** *In the two-category example, adaptation lowers*

expected loss by $\sqrt{K/2}$ before the audit and by $\sqrt{K(1/2+2\delta^2)}$ after a perfectly informative audit. The allocation value of the audit is therefore

$$\sqrt{K}\left(\sqrt{\frac{1}{2}+2\delta^2}-\sqrt{\frac{1}{2}}\right), \tag{42}$$

which is weakly increasing in δ and strictly increasing for $\delta > 0$.

The interpretation of Proposition 6 is that exposure heterogeneity creates a return to local information. When $\delta = 0$, the benchmark weights already match deployment, and the audit does not redirect spending. As δ rises, the benchmark becomes a less useful guide for adaptation. The audit does not improve the engineering technology. It improves the match between engineering effort and user exposure.

The selection effect can be seen in the same two-category environment. Suppose that the first category has greater residual loss than the second. Before an audit, the planner evaluates rollout using the expected deployment mix. After an audit, the planner can proceed when the low-loss category receives greater exposure and decline or delay rollout when the high-loss category receives greater exposure. The value of this option is largest when the possible deployment environments lie on opposite sides of the rollout threshold.

5.2 What an audit measures

An audit need not estimate a single distribution for all time. Its purpose is narrower: it should provide information that changes an imminent investment or rollout decision. A customer-service organisation might sample historical tickets and score the model on the resulting categories. A legal team might classify the types of contracts that occupy staff time. A software group might inspect the languages, repositories, and maintenance tasks that generate requests. In each case, the useful object is not only an average score. It is the joint distribution of exposure and local loss.

Audit design involves a trade-off between breadth and precision. A broad audit can identify omitted categories and reveal whether a public benchmark is poorly matched to local use. A focused audit can estimate exposure and loss more precisely in categories near an investment or verification threshold. The correct design depends on the decision. When the organisation is deciding whether to roll out at all, broad coverage is valuable. When rollout is likely, and the main question is where to adapt, precision in high-exposure categories becomes more important.

An audit can also be staged. An initial pilot estimates coarse exposure weights. The planner then adapts the model, retrieval system, or review policy. A second audit measures the residual tail. This sequence mirrors the portfolio problem in Section 4:

broad information can direct broad investment, while later information can direct targeted repair and verification.

5.3 Audits and broad scale

The value of an audit depends on the type of investment under consideration. Suppose that a scale investment s lowers loss by the same amount $h(s)$ in every category, does not hit a zero-loss floor over the relevant range, and has cost $C(s)$. For the moment, hold the rollout decision fixed and suppose that the scale budget is separate from the adaptation budget. The deployment-weighted gross gain from scale is

$$\sum_{i=1}^{n} d_i h(s) = h(s). \tag{43}$$

Uniform scale does not require knowledge of the deployment distribution because the weights sum to one.

Under these assumptions, subtracting cost from (43) gives the scale choice

$$\max_{s} \; h(s) - C(s), \tag{44}$$

which is independent of deployment weights. By contrast, the targeted-adaptation conditions in Section 2 depend directly on deployment weights. This is a uniform-versus-targeted distinction. An audit has no direct allocation value for an improvement that benefits every category equally. It matters for improvements whose returns vary across categories. This observation also clarifies why broad scale and deployment audits can coexist. A developer can rationally invest in broad capability before learning every use case. A deploying organisation still has reason to audit its workflow before allocating local adaptation resources.

The qualifications matter. If rollout is uncertain, an audit can change whether broad scale is worth purchasing for a particular environment. If scale and adaptation compete for the same budget, an audit can change the opportunity cost of scale by revealing attractive local repairs. The additive no-floor assumption also matters. If scale instead lowers losses proportionally, so that residual loss becomes $(1 - \theta(s))v_i$, the gross gain is $\theta(s)\mu_D$ and depends on deployment weights. If an additive improvement hits zero-loss floors in some categories, the gain is $\sum_i d_i \min\{h(s), v_i\}$ and again depends on exposure. The calculation isolates a useful benchmark: exposure information is most directly valuable when investment returns are uneven.

5.4 Verification after rollout

An audit improves ex ante choices but need not eliminate local risk. We next consider verification. Suppose that, after a task arrives, verification reveals the task-level loss before the organisation relies on the AI output. For a task in category i, this loss is the random variable V_i introduced in (1). The deployment expectation $\mathbb{E}_D$ first draws a category using deployment weights and then draws the task-level loss within that category. If realised loss exceeds the tolerance q, the organisation uses the outside option. Let $\bar{c}$ denote the per-task cost of verification. This is a perfect verification benchmark. With noisy verification, the same logic applies to posterior expected payoffs conditional on the verifier's signal.

Without verification, forced use generates expected payoff $q - \mu_D$. With verification, the organisation avoids the negative tail. The gross value of this safeguard relative to forced use is

$$\mathbb{E}_D\left[(V_i - q)_+\right]. \tag{45}$$

This expression is the expected overshoot of task-level loss above the tolerance threshold. It can be positive even when every category mean satisfies $v_i \leq q$, because verification acts on the within-category tail. If deployment itself is optional, the no-verification baseline is instead $\max\{q - \mu_D, 0\}$.

Proposition 7 (Verification)**.** *Relative to a forced-use baseline, blanket verification is worthwhile if and only if*

$$\bar{c} < \mathbb{E}_D\left[(V_i - q)_+\right]. \tag{46}$$

Relative to a baseline in which the organisation can choose not to deploy, blanket verification is worthwhile if and only if

$$\mathbb{E}_D\left[(q - V_i)_+\right] - \bar{c} > \max\{q - \mu_D, 0\}. \tag{47}$$

When $q \geq \mu_D$, condition (47) *reduces to* (46)*. If verification capacity is limited, it should be allocated first to categories with the largest conditional expected overshoot per unit of verification cost, $\mathbb{E}[(V_i - q)_+ \mid i]/c_i$, net of any category-specific review cost.*

The interpretation of Proposition 7 is an ex post screening effect. Verification is valuable when the residual task-level tail is sufficiently costly. Scale, regularity, and targeted repair can all lower this value by reducing residual loss. However, a system can be worthwhile on average and still have a tail large enough to justify verification.

5.5 Verification under uneven coverage

The uneven-coverage model gives the verification condition a geometric interpretation. Within a coverage region of length X, local loss is $XT(1 - T)$. For a given tolerance q,

tasks close to the anchors are relatively safe, while tasks near the midpoint are more likely to require manual review. In a short region, the entire interval can be safe. In a sufficiently long region, an interior set of tasks has loss above the threshold.

To see this, note that verification has positive gross value at a task whenever

$$XT(1-T) > q. \tag{48}$$

For a fixed region length X, the left-hand side is largest at the midpoint. Thus, a review policy should concentrate on the middle of long, sparse regions. This conclusion is the ex post analogue of Proposition 3: targeted coverage and targeted verification both place disproportionate attention on regions with poor support.

The two instruments differ in persistence. Adding an anchor changes the reliability of future tasks in a region. Verification incurs a cost each time a task is reviewed. Targeted repair is therefore attractive when a high-loss region is encountered frequently and can be improved at moderate cost. Verification is attractive when a residual failure mode is difficult to remove or when the organisation requires an immediate safeguard while adaptation is underway.

5.6 Audits and verification are distinct

It is useful to make the division of labour explicit. A deployment audit estimates which tasks will appear often enough to matter. It should therefore occur before adaptation and rollout. Verification observes whether a realised task is sufficiently risky to warrant abstention or manual review. It should therefore occur after the task arrives.

The instruments can be complements. An audit can reveal that a category is both frequent and difficult, making targeted verification worthwhile. They can also be substitutes. A sufficiently effective targeted repair reduces the overshoot loss in (45). The direction of this interaction depends on whether the audit mainly discovers an opportunity for repair or a persistent region in which repair is costly.

5.7 An integrated deployment-planning problem

The preceding results isolate the five margins so that their roles remain transparent. We now place them in a single workflow decision. Let $s \geq 0$ denote broad-scale spending, let $\rho \geq 0$ denote regularity spending, and let $a_i \geq 0$ denote targeted repair in category i. Let $A(Z) \geq 0$ denote the resource cost of acquiring audit signal Z, with $A(\varnothing) = 0$. Expenditure units are normalised so that a common reliability budget requires

$$A(Z) + s + \rho + \sum_{i=1}^{n} a_i \leq K. \tag{49}$$

Let L_i^0 denote baseline task-level loss in category i, with $\mathbb{E}[L_i^0 \mid i] = v_i$. Broad scale lowers task-level loss in every category by $h(s)$. Regularity is a common intervention with a fixed category loading $\omega_i \geq 0$, so it lowers category i by $\omega_i e(\rho)$. The loadings capture which categories are exposed to sparse or irregular coverage; the planner chooses the common regularity intensity ρ, not a separate regularity action in each category. Targeted repair lowers category i by $g_i(a_i)$. Residual task-level loss is

$$L_i(s, \rho, a_i) = \big(\underbrace{L_i^0}_{\text{baseline task loss}} - \underbrace{h(s)}_{\text{broad scale}} - \underbrace{\omega_i e(\rho)}_{\text{regularity}} - \underbrace{g_i(a_i)}_{\text{targeted repair}} \big)_+, \tag{50}$$

with $\ell_i(s, \rho, a_i) = \mathbb{E}[L_i(s, \rho, a_i) \mid i]$. Setting $s = \rho = 0$ and taking L_i^0 to be degenerate recovers the targeted-repair residual loss r_i in (14). The formulation is reduced-form and maps directly onto the instruments a deploying organisation controls. In the uneven-coverage model, broad scale raises λ, regularity lowers CV through a common compression of sparse regions, and targeted repair adds anchors in selected regions. In an organisational deployment, the corresponding interventions can include a model upgrade, retrieval design, data curation, workflow-specific adaptation, or a review protocol.

Suppose that the organisation purchases an audit signal Z before allocating its reliability budget. Let $\bar{d}_i(Z) = \mathbb{E}[d_i \mid Z]$ denote posterior expected workflow exposure. After a task arrives, the organisation can verify category i at operating cost $c_i \geq 0$ per task. Let $y_i \in \{0, 1\}$ indicate whether category i is verified. The category-level expected payoff is

$$u_i(s, \rho, a_i, y_i) = \underbrace{(1 - y_i)(q - \ell_i(s, \rho, a_i))}_{\text{reliance without review}} + \underbrace{y_i \left(\mathbb{E}[(q - L_i(s, \rho, a_i))_+ \mid i] - c_i\right)}_{\text{verification with outside option}}. \tag{51}$$

The first term applies when the organisation relies on the output without review. The second applies when verification allows the organisation to use the outside option after observing an unsafe task. The integrated value of an audit signal is

$$W(Z) = \mathbb{E}_Z \left[\max_{\substack{s, \rho, a_1, \ldots, a_n \geq 0; \\ y_1, \ldots, y_n \in \{0,1\}}} \left(\sum_{i=1}^{n} \bar{d}_i(Z) u_i(s, \rho, a_i, y_i) \right)_+ \right] \tag{52}$$

subject to (49). The positive-part operator captures rollout. The maximisation captures the reliability portfolio and the contingent verification policy.

Proposition 8 (Integrated deployment portfolio)**.** *For any audit realisation and any reliability portfolio, verification is optimal in category i with positive posterior exposure if and only*

if

$$c_i < \mathbb{E}\left[(L_i(s,\rho,a_i) - q)_+ \mid i\right]. \tag{53}$$

If signal Z' is more informative than signal Z in the sense of Blackwell (1953) and $A(Z') \leq A(Z)$, then $W(Z') \geq W(Z)$. At a differentiable local optimum with rollout, suppose the residual-loss distributions have no mass at 0 or q, and define

$$\psi_i = (1 - y_i)\Pr(L_i > 0 \mid i) + y_i \Pr(0 < L_i < q \mid i). \tag{54}$$

For every positive investment margin that is locally interior, the common-budget shadow value η satisfies

$$h'(s)\sum_{i=1}^{n} \bar{d}_i \psi_i = \eta, \tag{55}$$

$$e'(\rho)\sum_{i=1}^{n} \bar{d}_i \omega_i \psi_i = \eta, \tag{56}$$

$$\bar{d}_i \psi_i g_i'(a_i) = \eta. \tag{57}$$

When no loss floor binds and no category is screened out by verification, $\psi_i = 1$ for every category. The direct marginal return to broad additive scale is then $h'(s)$, while the return to regularity depends on the exposure-weighted loadings ω_i and the return to targeted repair depends on posterior workflow exposure category by category.

Proposition 8 connects the earlier margins. Better information can shift a common reliability budget even when it reveals no new engineering opportunity. If an audit raises posterior exposure in a weak category, it raises the direct return to targeted repair in that category. It also raises the return to regularity when the category has a high loading on sparse or irregular coverage. Broad scale remains attractive because it can improve the whole workflow, but the zero-loss floor and the verification threshold make its marginal value depend on the exposure-weighted mass of tasks that still benefit from a downward shift. A dollar assigned broadly also cannot be assigned to a local repair whose value becomes visible after the audit.

The verification rule adds a further tail boundary. Verification can be optimal even when the category mean ℓ_i is below q, provided the within-category expected overshoot is large enough. When residual loss is expensive to remove, the organisation can screen risky tasks after they arrive. When repair becomes sufficiently effective, the organisation can stop paying a recurrent review cost. Thus, an audit can change the operational policy as well as the capital budget: it can reveal where to improve the system, where to review its outputs, and whether rollout is worthwhile at all.

6 Developers, Organisations, and Measurement

The integrated planner is a useful starting point because it clarifies the available instruments. In practice, the decisions may be divided between a model developer and a deploying organisation. The developer observes broad benchmark results and serves many possible use cases. The organisation observes a narrower workflow and can acquire local information through a pilot or deployment audit. This division makes the design of public benchmark reports economically relevant.

6.1 A division of responsibility

A general-purpose developer is naturally placed to invest in scale and broad regularity. These investments can improve reliability across many deployment environments. A deploying organisation is naturally placed to estimate its own exposure weights, identify workflow-specific weak regions, and decide where retrieval support, adaptation, or review is warranted. The boundary is not sharp. A developer can collect product telemetry, while a large organisation can conduct substantial adaptation. The distinction is useful because it identifies who is likely to possess the relevant information.

This perspective also avoids an implausible account of organisational behaviour. An organisation need not ignore a known inspection paradox. It may simply lack a sufficiently accurate estimate of its deployment mix. A pilot programme can be valuable even when it reveals no new engineering facts about the underlying model. Its value can come from revealing which existing capabilities matter for a particular workflow.

6.2 What a public benchmark should disclose

A public benchmark must serve organisations with different workflows. This makes a single universal correction difficult. One organisation may deploy a coding assistant primarily for documentation, another for tests, and another for maintenance of legacy software. The appropriate exposure weights can differ even when the organisations evaluate the same underlying model.

The general framework yields a simple reporting result. Suppose that a public benchmark discloses its category weights $\mathbf{b} = (b_1, \ldots, b_n)$ and its aggregate loss $\mu_B = \mathbf{b} \cdot \mathbf{v}$, but not the category-level loss vector $\mathbf{v} = (v_1, \ldots, v_n)$. An organisation knows its own workflow weights $\mathbf{d} = (d_1, \ldots, d_n)$ and wishes to calculate workflow loss $\mu_D = \mathbf{d} \cdot \mathbf{v}$.

Proposition 9 (Public benchmark disclosure)**.** *Suppose that every category with positive workflow exposure appears in the public benchmark. For a specified workflow* $\mathbf{d}$*, the public*

benchmark mean μ_B identifies workflow loss μ_D for every possible category-level loss vector if and only if $\mathbf{d} = \mathbf{b}$. If the benchmark instead reports the category-level loss vector $\mathbf{v}$, the organisation can calculate $\mu_D = \mathbf{d} \cdot \mathbf{v}$ for any workflow supported by the benchmark. If some category has $d_i > 0$ and $b_i = 0$, neither aggregate reporting nor reweighting of reported benchmark categories identifies workflow loss in that category.

Proposition 9 identifies an information-design problem rather than a defect in public benchmarks. A conventional aggregate remains useful for comparing models on a common evaluation distribution. It is not a sufficient statistic for every workflow. When $d \neq b$, two systems can have the same public benchmark mean and different workflow losses. With a small perturbation, their public-benchmark ranking and workflow ranking can also differ. Category-level reporting allows an organisation to apply local weights after an audit, while coverage-omission reporting identifies cases in which the public evaluation suite must be expanded.

The proposition also explains why the aim is not to replace one public benchmark score with a single corrected score. A developer cannot generally anticipate every deployment distribution. A more useful disclosure provides a common aggregate for comparability and enough disaggregation for a deploying organisation to perform its own reweighting. Verification or abstention curves add a second dimension by showing how much residual tail loss can be managed operationally.

Granularity is nevertheless an economic design choice, not a free theorem. To see the trade-off, suppose primitive task types j are grouped into reporting cells $C \in \mathcal{P}$. A report that discloses only the benchmark-weighted cell mean $\bar{v}_C = \sum_{j \in C} b_j v_j / b_C$ allows an organisation to form $\hat{\mu}_D(\mathcal{P}) = \sum_{C \in \mathcal{P}} d_C \bar{v}_C$. The remaining within-cell bias is

$$\mu_D - \hat{\mu}_D(\mathcal{P}) = \sum_{C \in \mathcal{P}} \sum_{j \in C} \left(d_j - \frac{d_C b_j}{b_C} \right) v_j. \tag{58}$$

Refining the partition reduces this bias when workflow weights and benchmark weights differ inside a cell. It can also increase sampling variance, privacy risk, gaming incentives, or reporting cost. A stylised optimal reporting partition solves

$$\min_{\mathcal{P}} \left\{ \mathbb{E}\left[\left(\mu_D - \hat{\mu}_D(\mathcal{P}) \right)^2 \right] + \sum_{C \in \mathcal{P}} d_C^2 \sigma_C^2 + \kappa |\mathcal{P}| \right\}, \tag{59}$$

where σ_C^2 is the estimation variance of the reported cell mean and κ summarises disclosure costs. Category-level reporting is therefore optimal up to the point at which the reweighting value of a split is offset by noise and disclosure costs. Proposition 9 is the zero-noise, zero-disclosure-cost benchmark in this larger measurement-design problem.

Table 4: Existing benchmarks and workflow deployment

Benchmark	What it measures	What it contributes	What deployment still requires
MMLU (Hendrycks et al., 2021)	Multiple-choice accuracy across 57 academic and professional subjects	A broad screen of multitask knowledge and problem-solving capability	Workflow weights, stakes, and categories not represented by the subject mix
HELM (Liang et al., 2022)	Performance across a taxonomy of scenarios and multiple metrics	A transparent model of disaggregated reporting that recognises omitted scenarios and metric trade-offs	Organisation-specific scenario weights and operational thresholds
SWE-bench (Jimenez et al., 2024)	Resolution of real-world software issues drawn from GitHub repositories	A workflow-like evaluation of coding agents on repository-level tasks	The organisation's repository mix, languages, issue types, and review costs
TruthfulQA (Lin et al., 2022)	Truthfulness on questions designed to elicit common misconceptions	A targeted probe of an important failure mode	The frequency and consequences of related failures in the intended workflow
Chatbot Arena (Chiang et al., 2024)	Crowdsourced pairwise human preferences over model responses	A live evaluation platform whose prompt mix is closer to realised user demand than a fixed question suite	Reweighting for the organisation's users, tasks, stakes, and governance rules

6.3 How existing benchmarks support workflow decisions

Existing benchmarks illustrate why public evaluation and workflow deployment are related but distinct information problems. Table 4 describes five prominent examples. The examples are not intended to rank benchmark designs. Each benchmark answers a useful question. The issue is whether its reported objects are sufficient for a particular organisational decision.

MMLU covers 57 subjects, including mathematics, history, computer science, and law (Hendrycks et al., 2021). Its breadth makes it useful as a general capability screen. However, a subject average does not reveal how much weight a particular workflow should attach to each subject, nor does it identify tasks outside the benchmark taxonomy. HELM moves further towards the disclosure recommended by Proposition 9. It organises evaluation around a taxonomy of scenarios and multiple metrics, releases granular results, and explicitly recognises that a broad evaluation can still omit important scenarios (Liang et al., 2022). The remaining step is local: an organisation must decide which scenarios and metrics matter for its deployment.

SWE-bench provides a different kind of grounding. It tests whether a system can edit a codebase to resolve real-world GitHub issues, drawing the original task instances from issues and corresponding pull requests in Python repositories (Jimenez et al., 2024). This design is closer to a workflow than an abstract coding test. It does not eliminate the exposure problem. An organisation evaluating a coding agent still needs to know whether the repositories, languages, issue types, and verification costs represented in the benchmark resemble its own environment.

TruthfulQA and Chatbot Arena illustrate two additional evaluation roles. TruthfulQA contains 817 questions across 38 categories and is designed to elicit false answers associated with common misconceptions (Lin et al., 2022). It is naturally interpreted as

a targeted failure-mode probe rather than as an estimate of the average organisational workflow. Chatbot Arena uses crowdsourced questions and pairwise human preferences to evaluate models (Chiang et al., 2024). Its live prompt source moves evaluation closer to realised user exposure than a fixed suite, but the relevant user population and task mix remain deployment-specific. Other benchmarks can be classified in the same way. BIG-bench is a broad collaborative capability suite, while MT-Bench is a fixed multi-turn evaluation (Srivastava et al., 2023; Zheng et al., 2023). Their value depends on the decision they are asked to support.

None of these benchmarks literally samples one task from each latent coverage region in the model. Nor does the multiplier $1 + \mathrm{CV}^2$ provide a mechanical correction for their reported scores. The uneven-coverage model makes the information problem visible. Proposition 9 gives the practical implication: a public aggregate is a useful comparison object, but workflow deployment also requires disaggregated outcomes, local exposure weights, attention to omitted categories, and an operational policy for residual risk.

6.4 Empirical implications

The model yields four empirical implications. First, workflow-specific audits should redirect adaptation spending towards categories that are underweighted by public benchmarks but common in local use. Second, targeted retrieval and curation should have larger returns when deployment loss is concentrated in a small number of weak regions. Third, verification and abstention tools should be most valuable when rollout is worthwhile on average but local loss remains dispersed. Fourth, benchmark improvements should translate weakly into user satisfaction when the gains occur in categories with low deployment exposure.

These implications require measurement at the category level. In light of Proposition 9, an evaluator should report a conventional benchmark mean for comparability, category-level outcomes for local reweighting, a deployment-weighted mean when exposure weights are available, dispersion across task categories, a tail-loss measure such as worst-decile performance, and an abstention or verification curve. The purpose is not to replace a benchmark with a single corrected number. It shows which distribution is being measured and which decisions the measurement can support.

Table 5 keeps the reporting recommendation practical and close to current evaluation practice. The conventional mean remains useful because it supports comparisons across models and over time. Disaggregated results allow local reweighting. Dispersion indicates whether the mean is hiding unevenness. Deployment reweighting changes the average when a use case is known. Coverage-omission reporting identifies cases where reweighting alone is insufficient. Finally, a verification curve connects evaluation

Table 5: A minimal deployment reporting template

Reported object	Question answered	Data required
Conventional benchmark mean	How does the model perform on the evaluation distribution?	Existing benchmark results
Category-level outcomes	How can an organisation reweight the benchmark for its own workflow?	Outcomes by disclosed benchmark category
Category-level dispersion and worst-decile loss	How uneven is performance across measured categories?	Category-level outcomes
Deployment-weighted mean	How does the model perform on the intended workflow?	Category-level outcomes and exposure weights
Coverage omissions	Which deployment tasks are absent from the benchmark?	Workflow sample and benchmark map
Verification or abstention curve	How much residual loss can review remove at different review rates?	Task-level losses or review outcomes

to an operational safeguard.

6.5 From a public benchmark to a workflow decision

The model suggests a deployment sequence. The first step is to use public benchmarks to screen candidate systems and identify broad categories of weakness. The second step is to collect a workflow sample and estimate local exposure weights. The third step is to allocate adaptation spending based on the workflow distribution rather than on the public benchmark alone. The fourth step is to decide whether rollout is worthwhile after adaptation. The final step is to assign verification capacity to the residual tail.

This sequence is not intended as a universal procedure. A low-stakes use case can justify immediate rollout with lightweight monitoring. A high-stakes use case can require a broader audit and a more conservative verification policy. The point is that the relevant measurements differ across stages. A public benchmark is informative about general capability. It is not, by itself, a sufficient statistic for a workflow-specific investment and rollout decision.

The sequence also clarifies how a developer and organisation can exchange information. A developer can publish category-level performance, dispersion, and abstention curves. An organisation can contribute exposure information from its workflow. Neither party needs to observe every primitive in the model. The planner problem can be approached using measured category-level losses and exposure weights.

6.6 Interpreting the uneven-coverage model

The one-dimensional renewal model does substantive work. It shows how a mismatch can arise endogenously even when the benchmark and deployment environment concern the same underlying task line: a user is overexposed to large sparse regions because those regions occupy more of the line. It also delivers a transparent sufficient statistic, CV, for irregularity.

The general analysis does not depend on that interpretation. Task categories can be multidimensional, deployment weights can reflect demand rather than geometric size, and local loss can arise from mechanisms other than interpolation. The general condition remains the same: benchmark averages misstate deployment value when benchmark weights and deployment weights differ systematically in regions of uneven reliability.

6.7 Limitations

The model leaves several questions open. First, deployment weights are fixed in the main planner problem during the initial decision. The reduced-form endogenous-use extension in Section 2 shows how reliance can respond to quality and rules, but a dynamic adoption model would study how exposure evolves through repeated use and learning. Second, the planner is treated as a single decision maker. A developer, manager, and worker can have different objectives and different information. Third, most improvement margins are modelled as additively separable apart from the zero-loss floor and the common regularity loading. In many AI systems, fine-tuning, retrieval, data curation, or scaling can generate spillovers across categories or negative transfer. A more general technology would replace $g_i(a_i)$ with category losses $G_i(a_1, \ldots, a_n)$ and would value investment by deployment-weighted marginal effects $\sum_j d_j \partial G_j / \partial a_i$. Fourth, the uneven-coverage model treats coverage anchors as exogenous before investment. A richer production model would explain how data acquisition, retrieval, and model architecture jointly determine effective coverage. Fifth, the audit problem is represented by an abstract signal rather than by a full sampling model. Sample size, classification error, and estimation risk are important operational design choices.

These limitations point towards extensions rather than qualifications to the basic exposure logic. Endogenous use changes the distribution of deployment over time, but it does not eliminate the distinction between evaluation and exposure. Multiple decision makers can create agency problems, but each party still requires an estimate of the distribution relevant to their own choice. A richer production model can change the cost of scale, regularity, and targeted repair, but those margins remain economically distinct.

7 Conclusion

An AI benchmark and an AI deployment environment need not ask the same average question. A benchmark attaches evaluation weights to tasks. Users encounter deployment weights. When the benchmark-weighted covariance between local loss and relative deployment exposure is positive, benchmark loss understates the loss that users experience.

We have studied how this difference changes reliability investment. The general exposure identity does not depend on Brownian motion, Poisson coverage, or a one-dimensional task space. The renewal model supplies a concrete microfoundation: irregular coverage exposes users disproportionately to long regions, and Brownian-bridge interpolation makes those regions more error-prone on average. This yields the multiplier $1 + \mathrm{CV}^2$, with the Poisson factor of two as a special case.

The economic implication is a portfolio problem. Scale lowers the average loss. Regularity compresses the tail. Targeted coverage directs resources towards large weak regions. A deployment audit reveals where those investments matter for a particular workflow. Verification manages residual loss after rollout. When these margins are drawn on a common reliability budget, better exposure information can redirect spending towards local interventions whose value was hidden by the public benchmark. The relevant mix depends on the distribution of user exposure, not only on the average score reported by a benchmark.

This framework suggests a practical empirical agenda. Deployment logs can be used to estimate exposure weights. Pilot programmes can measure how these weights differ across organisations. Category-level outcomes can identify whether improvements are broad or concentrated and allow organisations to reweight a public benchmark for their own workflow. Verification data can reveal the remaining overshoot tail. A public benchmark mean remains useful for comparability, but it is not a sufficient statistic for workflow deployment. These measurements would permit a more direct comparison between benchmark progress and realised deployment value.

A Proofs

A.1 Proof of Proposition 1

Proof. Since $w_i = d_i/b_i$, we have $\mathbb{E}_B[w_i] = \sum_i b_i(d_i/b_i) = \sum_i d_i = 1$. Therefore,

$$\mathrm{Cov}_B(w_i, v_i) = \mathbb{E}_B[w_i v_i] - \mathbb{E}_B[w_i]\mathbb{E}_B[v_i] = \sum_i b_i \frac{d_i}{b_i} v_i - \mu_B = \mu_D - \mu_B.$$

The sign condition follows immediately: $\mu_D > \mu_B$ if and only if $\mathrm{Cov}_B(w_i, v_i) > 0$. □

A.2 Proof of Lemma 1

Proof. The stationary length-biased distribution is characterised by

$$\Pr(X^* \in A) = \frac{\mathbb{E}[X\mathbf{1}\{X \in A\}]}{\mathbb{E}[X]}$$

for every measurable set A. Equivalently, for every bounded measurable test function ϕ,

$$\mathbb{E}[\phi(X^*)] = \frac{\mathbb{E}[X\phi(X)]}{\mathbb{E}[X]}.$$

Taking $\phi(x) = x$ gives

$$\mathbb{E}[X^*] = \frac{\mathbb{E}[X^2]}{\mathbb{E}[X]}.$$

Finally, $\mathbb{E}[X^2] = \mathrm{Var}(X) + \mathbb{E}[X]^2 = \mathbb{E}[X]^2(1 + \mathrm{CV}^2)$. The same expression is obtained as the limit of finite renewal lines after ignoring boundary intervals, whose contribution vanishes as the domain grows. □

A.3 Proof of Proposition 2

Proof. Under region-uniform benchmark sampling, (17) implies

$$\mu_B = \frac{\mathbb{E}[X]}{6} = \frac{1}{6\lambda}.$$

Under deployment sampling, the encountered region length is X^*. Applying Lemma 1 gives

$$\mu_D = \frac{\mathbb{E}[X^*]}{6} = \frac{\mathbb{E}[X](1 + \mathrm{CV}^2)}{6} = \frac{1 + \mathrm{CV}^2}{6\lambda}.$$

Taking the ratio yields (22). □

A.4 Proof of Corollary 1

Proof. Under homogeneous Poisson coverage with intensity λ, region lengths are exponentially distributed with mean $1/\lambda$ and variance $1/\lambda^2$. Hence, CV $= 1$. Substituting into Proposition 2 gives the result. □

A.5 Proof of Proposition 3

Proof. Before the new anchor is added, the contribution of the original region to deployment loss is $X^2/(6L)$. Afterwards, the two new regions contribute

$$\frac{(\alpha X)^2 + ((1-\alpha)X)^2}{6L}.$$

The reduction in deployment loss is therefore

$$\frac{X^2 - \alpha^2 X^2 - (1-\alpha)^2 X^2}{6L} = \frac{\alpha(1-\alpha)X^2}{3L}.$$

The product $\alpha(1-\alpha)$ is maximised at $\alpha = 1/2$. □

A.6 Proof of Corollary 2

Proof. For one anchor, Proposition 3 shows that the gain from splitting a region of length X at α is $\alpha(1-\alpha)X^2/(3L)$. For any fixed X, this expression is maximised at $\alpha = 1/2$. Under a midpoint split, the gain is $X^2/(12L)$, which is increasing in X. Thus, one anchor should split the longest original region at its midpoint.

For multiple jointly planned anchors, fix an original region of length X_i and suppose m_i anchors are placed in it. If the resulting sublengths are $y_{i1}, \ldots, y_{i,m_i+1}$, their contribution to deployment loss is $(6L)^{-1}\sum_j y_{ij}^2$ subject to $\sum_j y_{ij} = X_i$. By convexity of y^2, or by the Cauchy–Schwarz inequality,

$$\sum_{j=1}^{m_i+1} y_{ij}^2 \geq \frac{X_i^2}{m_i+1},$$

with equality if and only if all sublengths are $X_i/(m_i+1)$. Hence, conditional on m_i, anchors should divide the original region i into equal subregions, and the planner must choose integer counts to minimise (35).

The loss contribution of region i with m_i assigned anchors is $X_i^2/[6L(m_i+1)]$. The marginal deployment-loss reduction from assigning one additional planned anchor to

that original region is therefore

$$\frac{X_i^2}{6L}\left(\frac{1}{m_i+1}-\frac{1}{m_i+2}\right)=\frac{X_i^2}{6L(m_i+1)(m_i+2)}.$$

These marginal gains are decreasing in m_i. The discrete allocation problem is therefore solved by selecting the M largest elements from the decreasing marginal-gain sequences for all original regions, which is equivalently implemented by the repeated marginal-gain rule in (36). □

A.7 Proof of Proposition 4

Proof. Before the new anchor is added, the average loss within the original region i is $X_i/6$. Afterwards, a uniformly drawn task from that original region falls in the left subregion with probability α and the right subregion with probability $1-\alpha$. Its average loss is therefore

$$\alpha\frac{\alpha X_i}{6}+(1-\alpha)\frac{(1-\alpha)X_i}{6}=\frac{(\alpha^2+(1-\alpha)^2)X_i}{6}.$$

The within-stratum reduction is $\alpha(1-\alpha)X_i/3$. Since the fixed benchmark gives weight $1/n$ to each original region, benchmark loss falls by $\alpha(1-\alpha)X_i/(3n)$. □

A.8 Proof of Proposition 6

Proof. Before the audit, symmetry implies $a_1=a_2=K/2$, so adaptation lowers expected loss by

$$\frac{1}{2}\sqrt{\frac{K}{2}}+\frac{1}{2}\sqrt{\frac{K}{2}}=\sqrt{\frac{K}{2}}.$$

After the audit, let $d_H=1/2+\delta$ and $d_L=1/2-\delta$. The planner solves

$$\max_{a_H+a_L\le K} d_H\sqrt{a_H}+d_L\sqrt{a_L}.$$

The first-order conditions imply $a_H/a_L=d_H^2/d_L^2$, which yields (41). Substitution gives an expected loss reduction of

$$\sqrt{K(d_H^2+d_L^2)}=\sqrt{K\left(\frac{1}{2}+2\delta^2\right)}.$$

Subtracting the no-audit loss reduction gives (42). The derivative with respect to δ is $2\sqrt{K}\delta/\sqrt{1/2+2\delta^2}$, which is zero at $\delta=0$ and positive for $\delta>0$. □

A.9 Proof of Proposition 5

Proof. A signal Z' that is more informative than Z in the sense of Blackwell allows the planner to reproduce any decision rule based on Z: after observing Z', the planner can apply the garbling that generates Z and then choose the corresponding adaptation allocation and rollout decision. The feasible set of decision rules under Z' therefore contains the feasible set under Z, which implies $V(Z') \geq V(Z)$. Strict inequality obtains exactly when some Z'-measurable decision rule yields strictly higher expected payoff than every payoff-maximising Z-measurable rule. A change that only selects among payoff-equivalent actions does not produce strict value. Subtracting the no-audit value $V(\varnothing)$ gives the audit condition. □

A.10 Proof of Proposition 7

Proof. Under forced use, the realised payoff for a task in category i is $q - V_i$. With verification and an outside option normalised to zero, the payoff before verification cost is $(q - V_i)_+$. The gross gain from verification relative to forced use is therefore

$$(q - V_i)_+ - (q - V_i) = (V_i - q)_+.$$

Taking the deployment expectation and subtracting $\bar{c}$ gives (46). If deployment is optional, the no-verification payoff is $\max\{q - \mu_D, 0\}$, while the verification payoff is $\mathbb{E}_D[(q - V_i)_+] - \bar{c}$. Comparing these two payoffs gives (47). When $q \geq \mu_D$, $\max\{q - \mu_D, 0\} = q - \mu_D$, and the identity

$$\mathbb{E}_D[(q - V_i)_+] - (q - \mu_D) = \mathbb{E}_D[(V_i - q)_+]$$

shows that the optional-rollout condition reduces to (46). With limited verification capacity, assigning verification to a category with a larger expected overshoot per unit of cost weakly dominates assigning it to a category with a smaller expected overshoot per unit of cost, all else equal. □

A.11 Proof of Proposition 8

Proof. For a category with positive posterior exposure, the expected payoff gain from verification relative to unverified reliance is

$$\mathbb{E}[(q - L_i)_+ \mid i] - c_i - (q - \ell_i) = \mathbb{E}[(L_i - q)_+ \mid i] - c_i,$$

where the equality follows from $\ell_i = \mathbb{E}[L_i \mid i]$. This proves (53). If Z' is more informative than Z in the Blackwell sense and is no more costly, a planner observing Z' can

reproduce any decision rule available after Z by applying the appropriate garbling and then selecting the same reliability portfolio, rollout decision, and verification policy within a weakly larger feasible set. Hence, $W(Z') \geq W(Z)$.

Consider a differentiable local optimum with rollout and no probability mass at the floor or verification threshold. A one-unit downward shift in residual task-level loss raises unverified payoff for tasks with $L_i > 0$ and raises verified payoff for tasks with $0 < L_i < q$. The marginal value of such a shift in category i is therefore $\bar{d}_i \psi_i$, with ψ_i defined in (54). The derivatives of the downward shift with respect to scale, regularity, and targeted repair are $h'(s)$, $\omega_i e'(\rho)$, and $g_i'(a_i)$, respectively. The first-order conditions for locally interior positive spending margins therefore equate their marginal payoff gains to the common-budget shadow value η, which gives (55)–(57). When no loss floor binds and no category is screened out, $\psi_i = 1$ for every category and $\sum_i \bar{d}_i = 1$, so (55) reduces to $h'(s) = \eta$. □

A.12 Proof of Proposition 9

Proof. If $\mathbf{d} = \mathbf{b}$, then workflow loss is the reported benchmark mean: $\mu_D = \mathbf{d} \cdot \mathbf{v} = \mathbf{b} \cdot \mathbf{v} = \mu_B$. Conversely, suppose that the benchmark mean identifies workflow loss for every possible category-level loss vector. Then any two loss vectors with the same benchmark mean must have the same workflow loss. Equivalently, for every perturbation $\mathbf{x}$ satisfying $\mathbf{b} \cdot \mathbf{x} = 0$, we must have $\mathbf{d} \cdot \mathbf{x} = 0$. Thus, the linear functional defined by $\mathbf{d}$ is proportional to the linear functional defined by $\mathbf{b}$. Since both weight vectors sum to one, the proportionality constant is one and $\mathbf{d} = \mathbf{b}$. The argument can be restricted to non-negative loss vectors by applying sufficiently small positive and negative perturbations around a strictly positive loss vector.

If the category-level vector $\mathbf{v}$ is reported, the organisation directly calculates $\mu_D = \mathbf{d} \cdot \mathbf{v}$ for any workflow supported by the benchmark. If some category has $d_i > 0$ and $b_i = 0$, the public benchmark contains no category-level performance observation for that workflow component. Reweighting the reported categories cannot recover the missing loss. □

B Further Discussion of the Uneven-Coverage Model

B.1 A stationary local formulation

The main text uses a long task line to keep the exposition concrete. An equivalent local formulation places a representative deployment task at the origin and studies the nearest anchors on each side. The origin is a coordinate normalisation, not an anchor at which the truth is known. In the Poisson case, the left and right distances from the task

to the nearest anchors are independent exponential random variables with intensity λ. Their sum is Gamma distributed with shape two and mean $2/\lambda$. This is the deployment version of the inspection paradox.

The distinction matters because a two-sided Brownian motion is often normalised by setting its value at the origin to zero. Such a normalisation should not be combined with a claim that the model is uncertain about the truth at that same known point. The large-interval presentation avoids that ambiguity.

B.2 Beyond Brownian-bridge loss

The Brownian bridge is sufficient but not necessary for the multiplier in Proposition 2. Suppose instead that the average loss in a region of length X is ζX for a constant $\zeta > 0$. Region-uniform benchmark loss is then $\zeta\mathbb{E}[X]$, while deployment loss is $\zeta\mathbb{E}[X^*]$. Their ratio remains $1 + \mathrm{CV}^2$.

More generally, if average loss is $\bar{v}(X)$, benchmark loss is $\mathbb{E}[\bar{v}(X)]$ and deployment loss is $\mathbb{E}[X\bar{v}(X)]/\mathbb{E}[X]$, as in (23). If $\bar{v}$ is nondecreasing, the covariance in (23) is nonnegative. To see this, let X' be an independent copy of X. Then

$$2\,\mathrm{Cov}(X, \bar{v}(X)) = \mathbb{E}\left[(X - X')\big(\bar{v}(X) - \bar{v}(X')\big)\right] \geq 0.$$

The inequality is strict when X is nondegenerate and $\bar{v}(X)$ is not almost surely constant. The linear specification is useful because it yields a transparent, exact multiplier.

B.3 More than one dimension

In a multidimensional task space, coverage anchors can generate cells rather than intervals. A randomly located task is more likely to fall in a large cell. This is the geometric analogue of length bias. However, the mapping from cell size to local loss depends on cell shape, the metric on task space, and the interpolation kernel. A multidimensional model, therefore, preserves the qualitative size-bias mechanism without preserving the one-dimensional constant.

B.4 From latent coverage regions to an observable workflow diagnostic

The coverage model is a microfoundation for uneven reliability, not an instruction to recover literal interpolation regions inside an AI system. A public benchmark need not sample one task from each latent region, and the multiplier $1 + \mathrm{CV}^2$ is not a mechanical correction for a named benchmark. An empirical implementation can work directly with the observable categories introduced in Section 2. Operational proxies for

Table 6: A constructed observable-category diagnostic

Category	b_i	d_i	v_i	$(d_i - b_i)v_i$
Routine drafting	0.50	0.20	0.05	−0.015
Legacy-system maintenance	0.30	0.50	0.18	0.036
Compliance review	0.20	0.30	0.10	0.010
Total	1.00	1.00		0.031

sparse support include nearest-neighbour distance in a representation space, retrieval similarity, low historical frequency, high reviewer disagreement, high abstention rates, or high pilot loss. These proxies are useful when they predict local loss and can be linked to workflow exposure.

For each category i, an evaluator records the benchmark weight b_i and a category-level loss estimate v_i. A workflow sample or pilot estimates deployment weight d_i. On the shared support $S = \{i : b_i > 0\}$, the relative-exposure ratio is $w_i = d_i / b_i$, and the contribution of category i to the benchmark–deployment wedge is

$$b_i(w_i - 1)v_i = (d_i - b_i)v_i. \tag{60}$$

If the categories in the sum cover the deployment support and the relevant v_i are observed, summing (60) recovers $\mu_D - \mu_B$. If instead some omitted category has $d_i > 0$ and $b_i = 0$, then a sum over benchmark-supported categories recovers only the reweighting component on shared support:

$$\sum_{i \in S} (d_i - b_i)v_i = \mu_D - \mu_B - \sum_{i \notin S} d_i v_i. \tag{61}$$

The omitted term requires separate evaluation rather than reweighting. Categories with positive shared-support contributions are those that make the public aggregate optimistic for the workflow. This diagnostic does not by itself determine the optimal intervention, because repair and verification costs can differ across categories. It identifies where exposure information is economically relevant and where a deploying organisation should investigate adaptation, abstention, or review.

Table 6 gives a constructed illustration with common support. It is not an estimate for any particular benchmark or organisation. The benchmark mean is 0.099, while the workflow-weighted mean is 0.130. The wedge of 0.031 is driven primarily by legacy-system maintenance, a category that is both relatively common in deployment and locally difficult. A workflow audit would therefore redirect attention towards that category even if the benchmark correctly reported every category-level loss.

The same calculation can be applied at different levels of granularity. A software organisation might use repositories, languages, and issue types; a legal team might use

contract types and review tasks; a customer-service unit might use ticket classes and escalation paths. Refining the categories is useful when benchmark and deployment losses differ within a coarse category. The empirical discipline is modest but important: report the evaluation distribution, estimate the workflow distribution, disclose local outcomes, and identify missing support before treating a public aggregate as a deployment statistic.

C A Minimal Reporting Template

The analysis suggests a reporting template that remains close to existing evaluation practice. First, report the conventional benchmark aggregate for comparability. Second, disclose category-level outcomes so that a deploying organisation can apply its own workflow weights. Third, report category-level dispersion, including at least one tail measure such as the worst decile, and identify material coverage omissions discovered in workflow audits. Fourth, when a target deployment environment is known, report an exposure-weighted aggregate using estimated deployment weights. Fifth, report the performance of a verification or abstention policy at a range of review rates.

These statistics answer different questions. The benchmark aggregate records performance on the evaluation distribution. Category-level outcomes permit local reweighting. Dispersion records unevenness. Coverage omissions identify tasks for which reweighting is not enough. The exposure-weighted aggregate estimates realised performance in a specified use case. The verification curve shows how much residual loss can be removed by ex post screening.

References

BARDHI, A. AND S. CALLANDER (2026): "Learning in a Correlated World," *Annual Review of Economics*.

BLACKWELL, D. (1953): "Equivalent Comparisons of Experiments," *Annals of Mathematical Statistics*, 24, 265–272.

BRESNAHAN, T. F. AND M. TRAJTENBERG (1995): "General Purpose Technologies "Engines of Growth"?" *Journal of Econometrics*, 65, 83–108.

BRYNJOLFSSON, E., D. LI, AND L. RAYMOND (2025): "Generative AI at Work," *Quarterly Journal of Economics*, 140, 889–942.

BURNELL, R., W. SCHELLAERT, J. BURDEN, T. D. ULLMAN, F. MARTINEZ-PLUMED, J. B. TENENBAUM, D. RUTAR, L. G. CHEKE, J. SOHL-DICKSTEIN, M. MITCHELL, D. KIELA, M. SHANAHAN, E. M. VOORHEES, A. G. COHN, J. Z. LEIBO, AND J. HERNANDEZ-ORALLO (2023): "Rethink Reporting of Evaluation Results in AI," *Science*, 380, 136–138.

CALLANDER, S. (2011): "Searching and Learning by Trial and Error," *American Economic Review*, 101, 2277–2308.

CARNEHL, C. AND J. SCHNEIDER (2025): "A Quest for Knowledge," *Econometrica*, 93, 623–659.

CHIANG, W.-L., L. ZHENG, Y. SHENG, A. N. ANGELOPOULOS, T. LI, D. LI, H. ZHANG, B. ZHU, M. JORDAN, J. E. GONZALEZ, AND I. STOICA (2024): "Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference," *arXiv preprint arXiv:2403.04132*.

DELL'ACQUA, F., E. MCFOWLAND III, E. MOLLICK, H. LIFSHITZ-ASSAF, K. C. KELLOGG, S. RAJENDRAN, L. KRAYER, F. CANDELON, AND K. R. LAKHANI (2026): "Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality," *Organization Science*.

GOODHUE, D. L. AND R. L. THOMPSON (1995): "Task–Technology Fit and Individual Performance," *MIS Quarterly*, 19, 213–236.

GUO, C., G. PLEISS, Y. SUN, AND K. Q. WEINBERGER (2017): "On Calibration of Modern Neural Networks," in *Proceedings of the 34th International Conference on Machine Learning*.

HENDRYCKS, D., C. BURNS, S. BASART, A. ZOU, M. MAZEIKA, D. SONG, AND J. STEINHARDT (2021): "Measuring Massive Multitask Language Understanding," in *International Conference on Learning Representations*.

JIMENEZ, C. E., J. YANG, A. WETTIG, S. YAO, K. PEI, O. PRESS, AND K. R. NARASIMHAN (2024): "SWE-bench: Can Language Models Resolve Real-World GitHub Issues?" in *International Conference on Learning Representations*.

KARATZAS, I. AND S. E. SHREVE (1991): *Brownian Motion and Stochastic Calculus*, Springer, 2 ed.

KOH, P. W., S. SAGAWA, H. MARKLUND, S. M. XIE, M. ZHANG, A. BALSUBRAMANI, W. HU, M. YASUNAGA, R. L. PHILLIPS, I. GAO, T. LEE, E. DAVID, I. STAVNESS, W. GUO, B. EARNSHAW, I. HAQUE, S. M. BEERY, J. LESKOVEC, A. KUNDAJE, E. PIERSON, S. LEVINE, C. FINN, AND P. LIANG (2021): "WILDS: A Benchmark of in-the-Wild Distribution Shifts," in *Proceedings of the 38th International Conference on Machine Learning*, vol. 139 of *Proceedings of Machine Learning Research*, 5637–5664.

LAI, V., C. CHEN, A. SMITH-RENNER, Q. V. LIAO, AND C. TAN (2023): "Towards a Science of Human–AI Decision Making: An Overview of Design Space in Empirical Human-Subject Studies," in *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, 1369–1385.

LIANG, P., R. BOMMASANI, T. LEE, ET AL. (2022): "Holistic Evaluation of Language Models," *arXiv preprint arXiv:2211.09110*.

LIN, S., J. HILTON, AND O. EVANS (2022): "TruthfulQA: Measuring How Models Mimic Human Falsehoods," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 3214–3252.

NOY, S. AND W. ZHANG (2023): "Experimental Evidence on the Productivity Effects of Generative Artificial Intelligence," Tech. rep., Massachusetts Institute of Technology.

ROSENBERG, N. (1982): "Learning by Using," in *Inside the Black Box: Technology and Economics*, Cambridge University Press.

ROSS, S. M. (1996): *Stochastic Processes*, Wiley, 2 ed.

SHIMODAIRA, H. (2000): "Improving Predictive Inference under Covariate Shift by Weighting the Log-Likelihood Function," *Journal of Statistical Planning and Inference*, 90, 227–244.

SRIVASTAVA, A. ET AL. (2023): "Beyond the Imitation Game: Quantifying and Extrapolating the Capabilities of Language Models," *Transactions on Machine Learning Research*.

SUGIYAMA, M., M. KRAULEDAT, AND K.-R. MÜLLER (2007): "Covariate Shift Adaptation by Importance Weighted Cross Validation," *Journal of Machine Learning Research*, 8, 985–1005.

ZHENG, L., W.-L. CHIANG, Y. SHENG, S. ZHUANG, Z. WU, Y. ZHUANG, Z. LIN, Z. LI, D. LI, E. P. XING, H. ZHANG, J. E. GONZALEZ, AND I. STOICA (2023): "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena," *arXiv preprint arXiv:2306.05685*.