AGO AI Quality Gate: Evidence-First Release Decisions for Retrieval-Augmented Generation

Giulio Zeloni, Enrico Lo Conte,
Salvatore Rionero, Giuseppe Santoro,
Alessandro Rastelli, and Fabio Sorrentino

Protom Group S.p.A.

Abstract. Enterprises adopting retrieval-augmented generation (RAG) face a recurring operational decision: promote, revise, or block a system version. The evidence is incomplete and the metrics come from fallible LLM judges. We report on AGO AI Quality Gate (AGO), an evidence-first quality-gate framework deployed in industrial RAG assessment engagements. AGO integrates four key components: a four-state decision model that treats missing data and judge errors as explicit outcomes; layered scoring combining deterministic checks, local guardrails, and structured LLM evaluation; a stratified beta-binomial gate that quantifies regression risk probabilistically; and a mandatory meta-evaluation protocol to validate the LLM judge before it influences decisions. Since engagement data is proprietary, we evaluate the judge layer on RAG-Bench, a public benchmark of 100k annotated RAG traces across 12 datasets. On identical stratified test samples (N=1200 per judge), a low-cost judge (gpt-4.1-nano) detects non-adherent answers barely above chance (AUROC 0.603 [0.570, 0.634]), despite producing flawless protocol output, while gpt-4o reaches 0.783 [0.756, 0.807]—yet its per-domain performance still ranges from 0.62 to 0.88. A fixed-seed gate study spanning regression, no change, and improvement quantifies unsafe promotion, false-alarm cost, and improvement throughput. Under regression, the decision-grade profile reduces unsafe promotion to $22.2\% - 35.1\%$, against $29.3\% - 41.8\%$ for a naive gate. These results support the design choices that judge quality must be measured per engagement and that point estimates alone are not a release decision.

Keywords: Retrieval-Augmented Generation · LLM-as-a-Judge · Quality Gates · Evaluation · Uncertainty Quantification

1 Introduction

Retrieval-augmented generation (RAG) [12] is now a standard architecture for grounding large language models (LLMs) in private document collections; deployments range from customer support and compliance to healthcare and finance [8]. Grounding mitigates but does not eliminate hallucination: answers may contradict retrieved evidence, cite the wrong source, or fabricate content

when retrieval fails [10]. In an enterprise setting this becomes a concrete, recurring decision about whether a given version can be released, or it needs additional reviews.

This paper reports on AGO AI Quality Gate (AGO), a self-hosted quality-gate platform built to support that decision. AGO is deployed at an IT consulting firm and used in RAG assessment engagements with enterprise clients. Engagement constraints shaped the design: client data cannot leave the premises; observability is often incomplete; evaluation datasets are small (tens to a few hundred cases) and stratified over imbalanced categories; and consultants must be able to defend every automated decision they sign.

A rich line of work addresses RAG *measurement*. Frameworks such as RAGAs [6] and ARES [15] decompose quality into retrieval-side and generation-side components scored by an LLM judge [17,13]. Benchmarks such as RGB [2] and RAGBench [7] assess the evaluators themselves. Deploying these components inside an accountable gate exposed three gaps between scoring and deciding. First, *evidence is routinely incomplete*: traces lack contexts, providers fail, judges return malformed output; averaging over whatever is available conflates "measured as bad" with "not measured". Second, *point estimates overstate certainty* on the small, stratified evaluation sets that real engagements afford. Third, *the measuring instrument is unvalidated*: LLM judges exhibit biases and task-dependent reliability [16,1], so agreement measured on one domain does not transfer to another.

The contributions of this work are fourfold and transferable to any organisation operating RAG gates.

First, evidence-first decision semantics deployed in production: a four-state decision lattice (promote, manual review, block, not evaluable) with per-metric decision modes and explicit treatment of missing evidence and judge protocol errors (Sect. 3.3). Second, a stratified beta-binomial gate: per-stratum credible intervals and a probabilistic regression risk $P(p_B < p_A - \delta)$ replace point estimates (Sect. 3.4). Third, per-engagement judge validation: a meta-evaluation protocol with balanced golden sets and threshold-free agreement statistics, mandatory before judge scores may influence decisions (Sect. 3.5). Fourth, a public evaluation at two system boundaries: judge-layer validation on RAGBench (Sect. 5.1), operating characteristics of the production statistical gate under regression, no change, and improvement (Sect. 5.2).

Each component builds on established work. The contribution of AGO is the decision layer that makes missing evidence, judge protocol errors, statistical uncertainty, and per-engagement judge validity explicit inputs to a release decision, not hidden assumptions behind a score.

The paper is organized as follows. Sect. 2 reviews related work on RAG evaluation and LLM-as-a-judge reliability. Sect. 3 presents the AGO framework: evidence model, layered scoring, decision semantics, statistical gate, and judge meta-evaluation. Sect. 4 describes the deployed platform and the engagement workflow. Sect. 5 reports the evaluation: judge-layer validation on RAGBench and operating characteristics of the statistical gate. Sect. 6 concludes.

2 Related Work

RAG evaluation frameworks. RAGAs [6] introduced reference-free, LLM-scored metrics for RAG (faithfulness, answer relevance, context precision and recall). ARES [15] trains lightweight judges on synthetic data and calibrates them with prediction-powered inference over a small human-labelled set. It targets evaluator accuracy on benchmarks; AGO instead embeds evaluator validation inside an end-to-end decision pipeline with explicit missing-evidence semantics. Open-source tooling popularised the RAG triad of context relevance, adherence, and answer relevance, which we adopt as the judge-facing decomposition. RAGAs, ARES, and TruLens provide scores or evaluator predictions rather than the evidence-aware release semantics studied here; direct comparison of release decisions would therefore require imposing a common decision policy. Toolkits such as NeMo Guardrails [14] provide programmable input and output rails; AGO incorporates comparable local guardrails as one metric provider among several, subject to the same decision semantics.

Benchmarks for RAG and its evaluators. RGB [2] probes LLM behaviour under noisy or counterfactual retrieval. RAGBench [7] releases 100k (question, documents, response) traces from 12 datasets across five industry domains, annotated with the TRACe schema: continuous *relevance*, *utilization*, and *completeness* scores and a binary *adherence* label. The labels are produced by LLM annotators calibrated against human annotations; we therefore refer to them as *reference labels* rather than ground truth. RAGBench frames evaluator quality as a supervised prediction problem (RMSE for continuous targets, AUROC for adherence), letting heterogeneous evaluators be compared on equal footing; we adopt this protocol.

LLM-as-a-judge reliability. LLM judges correlate with human preferences on some tasks [17,13] but exhibit position and verbosity biases [16], and large-scale studies report agreement varying substantially across tasks and domains [1]. This non-transferability motivates our per-engagement validation: rather than certifying a judge model once, AGO re-measures judge–human agreement for every deployment domain, using classical agreement statistics [4,11], imbalance-robust summaries [3], and bootstrap uncertainty [5]. Existing tooling scores answers. Converting scores into accountable release decisions under incomplete evidence is the gap deployed systems face, and the gap this work addresses.

3 The AGO AI Quality Gate Framework

3.1 Setting and Evidence Model

An *evaluation case* $x = (q, E, K^*, T, S, P)$ consists of a question q, an optional expected behaviour E (reference answer, expected terms T, expected sources S), optional retrieval requirements K^*, and forbidden phrases P. A versioned dataset $D = \{x_i\}_{i=1}^{n}$ is assembled from observed production traces; each case

carries a category $c(x)$ and a criticality flag. Invoking the system under test on x_i yields an answer a_i and retrieved contexts K_i (possibly empty).

A set of *metric providers* maps (x_i, a_i, K_i) to metric results: each result m carries a name, an optional numeric value v, an optional threshold θ, a boolean pass flag, and the emitting provider. All thresholds, synonym policies, and judge instructions are versioned in an *evaluation profile*, never hard-coded: every decision is reproducible from (dataset snapshot, profile snapshot, code version). The framework distinguishes four evidence types throughout: *observed*, *declared*, *inferred* (flagged as such), and *missing*. Missing evidence must surface as an explicit gap with an impact statement, never as a default value.

3.2 Layered Scoring

Deterministic checks. The first layer is local, cheap, and exactly reproducible: answer presence, presence of required retrieved contexts, matching of expected sources and document versions, detection of forbidden phrases, and matching of expected terms with normalisation of Unicode, case, and accents plus profile-versioned synonym groups. Term matching passes when the matched fraction reaches a profile-defined ratio. These checks provide a judge-independent baseline and catch structural failures early.

Local guardrails. A rule engine inspired by production rail systems [14] evaluates pre- and post-conditions (PII, secrets, prompt injection, forbidden phrases, JSON and link validity) with per-rule actions $\{\mathsf{block}, \mathsf{warn}, \mathsf{log}\}$, timeouts, and fail-open/fail-closed behaviour. Every rule execution emits a metric result, so a guardrail timeout is visible evidence rather than a silent skip.

LLM-judge RAG triad. The third layer scores each case on the RAG triad of context relevance c, adherence g, and answer relevance r through a single constrained JSON call to an exchangeable, OpenAI-compatible judge endpoint at temperature 0. A metric passes if its score meets the profile threshold θ. Two protocol rules are central. First, scores are accepted only within a small tolerance of the declared range: values in $[-0.01, 1.01]$ are clamped to $[0, 1]$; non-numeric, non-finite, or genuinely out-of-range values are *provider protocol errors*. The metric is then marked failed with an explanatory comment and the case is routed to manual review. Malformed judge output is never coerced to a valid score. A silent zero is indistinguishable from a measured catastrophic failure, and a silent clamp could convert an invalid output into a confident pass. Second, adherence is scored only for cases that require retrieval; for purely conversational turns the dimension is undefined and omitted rather than imputed.

At run level, the primary grounding measure is the *adherence rate*: the fraction of evaluated cases that passed the adherence criterion. Stakeholders thus receive an interpretable proportion, not an average of uncalibrated judge scores; the raw scores remain per-case diagnostics.

3.3 Decision Semantics

Let $M(x)$ be the metric results for case x and $F(x) = \{m \in M(x) : \neg m.\text{passed}\}$ the failed subset. The profile assigns each metric a decision mode $\mu(m) \in \{\textsf{observe}, \textsf{review}, \textsf{block}\}$; observe metrics are reported but never influence decisions. A fixed, documented set C of *critical metric identifiers* (missing required contexts, failed source or version matching, forbidden phrases, PII, secret, or injection findings, output-format violations) escalates independently of mode. The item decision is

$$\text{dec}(x) = \begin{cases} \textsf{not evaluable} & M(x) = \emptyset, \\ \textsf{block} & \exists\, m \in F(x) : \mu(m) = \textsf{block} \ \vee\ \text{name}(m) \in C, \\ \textsf{manual review} & F(x) \setminus \{m : \mu(m) = \textsf{observe}\} \neq \emptyset, \\ \textsf{promote} & \text{otherwise}, \end{cases} \tag{1}$$

and run-level aggregation is conservative:

$$\text{dec}(D) = \begin{cases} \textsf{block} & \exists\, i : \text{dec}(x_i) = \textsf{block}, \\ \textsf{not evaluable} & \forall\, i : \text{dec}(x_i) = \textsf{not evaluable}, \\ \textsf{manual review} & \exists\, i : \text{dec}(x_i) \in \{\textsf{manual review}, \textsf{not evaluable}\}, \\ \textsf{promote} & \text{otherwise}. \end{cases} \tag{2}$$

The set C is a fixed system safety floor: membership in C takes precedence over a per-metric profile mode and therefore blocks even if that metric was mistakenly configured as observe-only. Outside C, the versioned evaluation profile determines whether a failed metric is blocking, review-only, or observational; this precedence is part of the decision semantics, not a profile-dependent risk preference. In deployment, profile-level blocking is reserved for critical violations (e.g. PII leakage), and every non-promote outcome carries machine-readable failure reasons and is routed to consultant review. Two properties follow. *No implicit pass*: absent evidence can only produce not evaluable or manual review, never promote. Human-in-the-loop *by construction*: the gate is an escalation mechanism, not an autonomous verdict.

3.4 The Stratified Hierarchical Beta-Binomial Gate

AGO models release quality at the operational levels the gate acts on. Item decisions are grouped into engagement-defined strata. Each stratum receives a beta-binomial posterior. Run-level quality is a weighted posterior aggregate over strata, and baseline and candidate runs are compared at the posterior level. The hierarchy is operational - item $\rightarrow$ stratum $\rightarrow$ run $\rightarrow$ comparison - not a global partial-pooling Bayesian model. The implementation is deliberately simple, auditable, and profile-configurable.

Point pass-rates on small, stratified evaluation sets are misleading. Per category c, we model the item outcome $X_i = \mathbf{1}[\text{dec}(x_i) = \textsf{promote}]$ as Bernoulli with

rate p_c under a conjugate prior $p_c \sim \mathrm{Beta}(\alpha_0, \beta_0)$ (default $\mathrm{Beta}(1,1)$) [9]; observing k_c successes among n_c cases gives the posterior $\mathrm{Beta}(\alpha_0 + k_c,\ \beta_0 + n_c - k_c)$, reported with equal-tailed 95% credible intervals per stratum. The run summary is a weighted posterior mean $\bar{p} = \sum_c w_c\, \mathbb{E}[p_c \mid \text{data}]$ with weights defaulting to dataset coverage $w_c = n_c/N$. Since evaluation sets are often deliberately enriched with critical cases, coverage weights reflect the *evaluation design*, not production traffic; profiles may therefore supply explicit target weights (traffic shares or business criticality) when a production-facing estimate is required, and the report states which weighting is in effect.

For version comparison, let A (baseline) and B (candidate) be runs on the same dataset and profile. Drawing S Monte Carlo samples of the stratified weighted rate for each run (fixed, recorded seeds), the regression risk is

$$\rho = P\big(p_B < p_A - \delta \mid \text{data}\big), \tag{3}$$

with δ the minimal regression of practical relevance (default 0.03); the statistical gate returns block if $\rho \geq \tau_b$, manual review if $\rho \geq \tau_r$ (defaults 0.8, 0.5), and promote otherwise. It supports the operational decision of Eq. (2) and never overrides blocking evidence. The two runs are modelled as independent. When they share items, positive correlation means independence *overestimates* the variance of the difference, so the gate errs toward manual review and block. This approximation is conservative by design: an unnecessary manual review costs consultant time, while a silently promoted regression costs a production incident. This conservative treatment is intentional: when baseline and candidate evidence is too similar, the gate should expose the uncertainty and route the comparison to human review rather than silently promote a risky release. The uninformative default prior is also deliberate. It confines the evidence about a candidate to the candidate's own run, rather than anchoring it to earlier versions, and keeps the audit trail free of hidden information. Engagements with defensible domain history may supply an informative prior through the versioned profile (α_0, β_0 are profile parameters). Likewise, δ is an engagement-level risk parameter, not a constant; stratum-specific δ is a natural profile extension. All hyperparameters live in the versioned profile and are reported with each run. Changing the release risk appetite means changing versioned thresholds, not the evidence model or hidden code constants. Sect. 5 evaluates judge behavior under the frozen meta-evaluation protocol and the statistical gate's operating characteristics under three controlled data-generating scenarios.

3.5 Per-Engagement Judge Meta-Evaluation

Judge–human agreement is task- and domain-dependent [1,16]; AGO therefore treats it as a quantity to re-measure per engagement. Before judge-based metrics may influence gate decisions for a new deployment domain, the following protocol must be executed and its report attached to the engagement record.

Golden set. A domain golden set with binary human labels per judge dimension: at least 30 cases as an absolute floor for a smoke-level check, and 100 or

more for decision-grade validation; both classes present for every dimension (minority class $\geq 30\%$); a deliberate share of borderline cases; independent labelling by two annotators with adjudication, and inter-annotator agreement reported as the ceiling against which judge agreement is read.

Statistics. Judge scores are binarised at the operating threshold θ and compared to labels: raw agreement; Cohen's κ [4,11], reported as *undefined* when expected agreement is 1 (single-class labels) rather than coerced; the Matthews correlation coefficient, informative under class imbalance [3]; threshold-free AUROC over raw scores; a threshold sweep, since θ is itself a profile parameter; and 95% bootstrap confidence intervals [5] throughout. A judge-dependent metric may run in block mode only when its validation report meets the profile's minimum lower-bound and maximum interval-width criteria. In practice an engagement starts with judge metrics in review-only mode and graduates them as the golden set matures. Validation evidence is bound to the exact judge configuration (model identifier, frozen rubric, operating threshold). Hosted judge models change without notice, so switching judge endpoint or model version invalidates the report and re-triggers the protocol.

4 Deployment

AGO is implemented as a self-hosted platform (API backend and web console) deployed at an IT consulting firm and in active use for RAG assessment engagements with enterprise clients. Deployment constraints shaped the architecture. Client data must not leave the premises: all storage, scoring, and reporting run inside the client or consultant perimeter. The only outbound call is the judge endpoint, exchangeable between a cloud API and a self-hosted OpenAI-compatible runtime for deployments that forbid egress. Observed traces arrive from heterogeneous observability stacks and are normalised into a provider-agnostic schema. Personally identifiable information and secrets are detected and redacted before a trace can become a dataset case. Evaluation datasets are curated before any gate run: duplicates, weak cases, coverage gaps, and unresolved privacy findings are surfaced, and blocking findings prevent the gate from running at all.

An engagement follows a repeatable workflow: import observed conversations; curate a versioned dataset; select or adapt an evaluation profile (thresholds, guardrail policy, judge policy, statistical policy-all versioned); execute the per-engagement judge meta-evaluation of Sect. 3.5; run the gate; and deliver a report. Every decision in the report carries its failure reasons, evidence types, per-stratum uncertainty, and gaps. Every run persists content-hashed snapshots of dataset and profile, so a decision remains auditable after either evolves. Non-promoted cases are reviewed by consultants; the gate escalates, humans decide.

5 Evaluation

We evaluate two claims at their appropriate boundaries. First, RAGBench tests the judge layer and the frozen meta-evaluation protocol; it is not an end-to-

end validation of AGO. Second, controlled simulation measures the statistical gate's unsafe-promotion risk, false-alarm cost, and improvement throughput. All gate-level experiments are local and make no external model or benchmark calls.

5.1 Judge-Layer Validation on RAGBench

Engagement data is proprietary, so we evaluate the judge layer on a public benchmark. RAGBench provides complete observed traces, so it plugs directly into AGO's evaluation contract without running any retrieval system. Each instance becomes a case with question, answer, and retrieved contexts. We ask: *how well do the framework's judge metrics agree with the benchmark's reference labels, how does this depend on the judge model, and what does it imply for gate operating points?*

Benchmark and Metric Mapping. The benchmark [7] is organised into 12 component datasets across five industry domains (listed in Table 3), with TRACe reference labels produced by LLM annotators calibrated against human annotations. We map our adherence score to the binary `adherence` label, compared by AUROC and by MCC/F1 at the deployed operating threshold $\theta = 0.8$. Our context-relevance score is holistic, whereas the TRACe `relevance` label is a fraction of relevant context spans; the definitions differ, so rank correlations (Spearman, Kendall) are the primary comparison and RMSE is reported only as supplementary material with this caveat. Answer relevance has no TRACe counterpart and is excluded; utilization and completeness are not natively produced by our judge and are left to future work.

Protocol. All reported numbers use stratified samples of 100 instances per component dataset, balanced by adherence label where possible, with fixed published seeds. These samples support controlled comparison, not population-level performance estimates. Sample manifests (exact instance identifiers) are persisted, so every judge scores identical instances. Judge responses are cached by (instance, model, rubric hash). The rubric was frozen before any test-split instance was scored; no prompt or threshold was adjusted afterwards. Judge model selection is part of the protocol: candidate judges (`gpt-4.1-nano`, `gpt-4o`; temperature 0) were compared on a *validation*-split sample of three datasets (N=300), with MCC at the deployed threshold as the pre-specified primary criterion. The selected judge and the low-cost judge were then run once on the *test* split of all 12 datasets (N=1200 per judge). All statistics carry 95% bootstrap confidence intervals (1,000 resamples, fixed seed). Judge protocol errors are excluded from agreement statistics and reported separately; none occurred in the reported runs.

Pre-registered sanity audits caught two pipeline defects before the reported runs: a payload bug silently capping the number of documents passed to the judge, and duplicate instance identifiers across component datasets. The latter forced aborting a first full-test attempt and discarding its partial results. Payload audits confirm that in all reported runs every document of every instance reached the judge.

Table 1. Judge selection on the validation split (3 datasets, N=300 per judge; 95% bootstrap CIs). MCC is computed at the deployed threshold θ=0.8.

Judge	AUROC adh.	MCC @ 0.8	Spearman rel.	Cost
gpt-4.1-nano	0.672 [0.609, 0.732]	0.284 [0.176, 0.391]	0.279 [0.163, 0.386]	$0.06
gpt-4o	0.787 [0.731, 0.837]	0.435 [0.329, 0.539]	0.217 [0.103, 0.330]	$1.36

Table 2. Test-split results over all 12 RAGBench datasets (N=1200 per judge, identical instances; 95% bootstrap CIs; overall micro aggregation).

Judge	AUROC adh.	MCC @ 0.8	Spearman rel.	Cost
gpt-4.1-nano	0.603 [0.570, 0.634]	0.170 [0.110, 0.221]	0.366 [0.313, 0.413]	$0.39
gpt-4o	0.783 [0.756, 0.807]	0.433 [0.386, 0.484]	0.298 [0.243, 0.351]	$9.42

Results.

Judge selection. Table 1 reports the validation comparison. By the pre-specified criterion (MCC at the operating threshold), gpt-4o was selected for the full test run. The same protocol makes judge cost-quality trade-offs directly measurable per engagement.

Judge quality dominates. Table 2 shows the headline result. On identical test instances, gpt-4o exceeds the low-cost judge by +0.180 AUROC and +0.263 MCC at the deployed threshold, with non-overlapping confidence intervals. The low-cost judge detects non-adherent answers only weakly above chance (AUROC 0.603). Its per-class score distributions overlap heavily; no threshold in a 0.1–0.9 sweep yields MCC above 0.17, so the deficiency is not a threshold artifact. Nothing in its operational output signals this: across all runs it produced zero protocol errors and well-formed, plausible explanations. An earlier pilot on three datasets (N=300) had yielded AUROC 0.563 [0.495, 0.631] for the same judge. That interval includes chance, which is why the framework mandates confidence intervals before conclusions are drawn.

Reliability is domain-dependent even for the strong judge. Table 3 disaggregates the selected judge. AUROC ranges from 0.620 (`cuad`, legal contracts) to 0.883 (`expertqa`), and at the deployed threshold MCC ranges from 0.040 (`emanual`) to 0.618 (`msmarco`): on some domains the production operating point is close to uninformative even for a frontier judge. This is the empirical core of the central design requirement: judge quality cannot be certified once and assumed elsewhere. It must be measured per engagement, with thresholds calibrated per domain -exactly what the versioned profiles and the meta-evaluation protocol implement. The relevance rank correlations are low and unstable across datasets (including negative values), consistent with the definitional mismatch noted in the metric mapping above; we treat this comparison as exploratory.

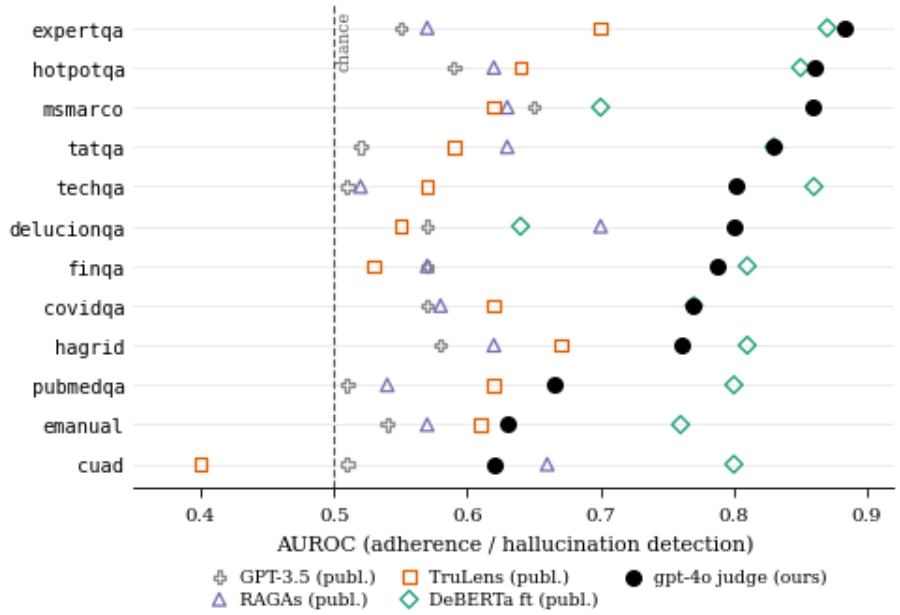

Fig. 1. Per-dataset adherence AUROC of the selected judge (gpt-4o, filled circles; stratified test samples, N=100 per dataset) alongside evaluator results published by the benchmark authors on full test splits (their Table 3 [7]). Dashed line: chance. Datasets sorted by gpt-4o AUROC; the comparison across protocols is indicative (Sect. 5.1). Per-dataset MCC and Spearman values appear in Table 3.

Positioning against published evaluators. The benchmark authors report AUROC for response-level hallucination detection (equivalently, adherence discrimination) on the full test splits for a zero-shot GPT-3.5 judge, RAGAs, TruLens, and a DeBERTa-large encoder fine-tuned on RAGBench (their Table 3 [7]): across the 12 component datasets, GPT-3.5 ranges 0.51–0.65 (mean 0.56), RAGAs 0.52–0.70 (mean 0.60), TruLens 0.40–0.70 (mean 0.59), and the fine-tuned DeBERTa 0.64–0.87 (mean 0.79). Under our protocol, the gpt-4o judge (per-dataset mean 0.77, micro 0.783) falls within the range reported for the fine-tuned encoder and above the zero-shot baselines, while our low-cost judge falls within the range of the earlier zero-shot evaluators. Published RAGBench baselines are included for orientation only, not as a protocol-identical comparison: the published numbers use full test splits and earlier GPT-3.5-backed evaluators, whereas our judge layer is evaluated on fixed stratified samples of 100 instances per dataset under a frozen judge protocol. The ordering nonetheless reinforces the central point: evaluator quality varies widely and must be measured, not assumed. These results support the judge-layer and meta-evaluation analysis under the tested benchmark conditions; they do not validate the full release gate.

Table 3. Per-dataset test results for the selected judge (gpt-4o, N=100 per dataset).

Dataset	AUROC (adherence)	MCC @ 0.8	Spearman (relevance)
`covidqa`	0.769	0.388	0.009
`cuad`	0.620	0.199	0.201
`delucionqa`	0.800	0.166	0.236
`emanual`	0.630	0.040	0.324
`expertqa`	0.883	0.572	0.400
`finqa`	0.787	0.446	0.233
`hagrid`	0.761	0.319	0.128
`hotpotqa`	0.860	0.530	−0.159
`msmarco`	0.859	0.618	0.363
`pubmedqa`	0.665	0.311	0.361
`tatqa`	0.829	0.563	−0.053
`techqa`	0.802	0.393	0.583

5.2 Operating Characteristics of the Statistical Gate

We exercise the same statistical-benchmark paths used by the product, `build statistical benchmark` and `compare statistical benchmarks`. Each synthetic run contains N binary item decisions ($N = 20$, 30, 50, 100, or 200), split across `general`, `domain specific`, and `critical` strata in proportions 0.50, 0.30, and 0.20. Their baseline pass probabilities are 0.90, 0.85, and 0.80. Candidate probabilities change by -0.05, 0, or $+0.05$ in every stratum, defining the *regressed*, *unchanged*, and *improved* scenarios. For each scenario and N, we run 1,000 replicates with data seed 20260709. Baseline and candidate observations are independent within each replicate. The beta-binomial gate uses a $\mathrm{Beta}(1, 1)$ prior, 95% credibility, minimum relevant regression $\delta = 0.03$, block threshold 0.80, 121 posterior samples, and independent posterior streams seeded at 4201 and 4202. Run-level posterior draws use the empirical stratum fractions induced by the stated allocation.

We compare a naive gate, which promotes when the observed candidate pass rate is at least the observed baseline pass rate minus δ, with two AGO profiles. `balanced default` has review threshold 0.50; `decision grade`, used in Table 4, has review threshold 0.40. Both profiles receive exactly the same item observations and posterior regression probability in each replicate, so their only controlled difference is the review threshold. The seed schedule was fixed in an earlier regression-only pilot of the same generator and reused unchanged when the two additional scenarios were added; it is not the result of a seed search. It does not create a paired Bayesian model: baseline and candidate item outcomes, and their posterior draws, remain independent.

In these synthetic scenarios, promotion of a regressed candidate is an unsafe promotion. Under no true change, manual review or block is a false alarm. Under true improvement, promotion measures release throughput. We report all three outcomes together to expose both the protection and the operational cost of each policy.

Table 4. Gate operating characteristics for the `decision grade` profile (rates in %).

N	Naive promote	AGO promote	AGO review	AGO block	Avg. ρ
Regressed					
20	41.8	33.0	47.3	19.7	0.532
30	37.4	35.1	41.3	23.6	0.538
50	38.1	34.5	40.8	24.7	0.546
100	38.4	29.9	42.5	27.6	0.571
200	29.3	22.2	41.6	36.2	0.638
Unchanged					
20	59.7	51.4	38.3	10.3	0.419
30	56.3	52.7	38.1	9.2	0.408
50	67.9	60.7	31.7	7.6	0.356
100	76.6	70.0	25.2	4.8	0.286
200	82.1	76.2	21.1	2.7	0.249
Improved					
20	76.7	71.5	24.4	4.1	0.297
30	82.3	78.9	19.6	1.5	0.245
50	89.7	85.1	13.5	1.4	0.176
100	97.7	95.9	4.0	0.1	0.081
200	99.8	99.4	0.6	0.0	0.028

Table 4 shows a consistent decision-grade reduction in unsafe promotion under true regression. Across $N = 20$–200, the naive gate promotes 29.3%–41.8% of regressed candidates, whereas AGO promotes 22.2%–35.1%; the reduction holds at every tested N. The protection is not free. Under no true change, the decision-grade profile promotes 51.4%–76.2% and sends 23.8%–48.6% to review or block. Under true improvement, it promotes 71.5%–99.4%, compared with 76.7%–99.8% for the naive rule. As N grows, the average posterior regression probability separates the scenarios: at $N = 200$ it is 0.638 for regression, 0.249 for no change, and 0.028 for improvement.

The balanced profile is reported in the accompanying artifact rather than hidden. Under regression it promotes 29.8%–43.9%, which can match or exceed the naive rate, while under no change it promotes 60.6%–83.0% with a 17.0%–39.4% false-alarm rate. Under improvement it promotes 77.4%–99.8%. Because the two profiles share all simulated evidence, the stronger regression protection of `decision grade` is attributable to its lower review threshold, not to different samples or a different posterior model. These operating characteristics describe policy trade-offs under the tested data-generating process.

6 Conclusion

AGO AI Quality Gate turns RAG evaluation from score production into an auditable release decision. The RAGBench study shows that judge reliability must

be measured under a frozen meta-evaluation protocol, not inferred from well-formed output. The three-scenario gate experiment exposes the release policy's operating characteristics: the decision-grade profile lowers unsafe promotion at every tested sample size, at an explicit false-alarm and throughput cost.

Deployment and evaluation yielded four operational findings. First, judges can fail silently: the low-cost judge produced flawless, plausibly explained output while discriminating barely above chance; only measurement against labelled references revealed it. Second, confidence intervals change conclusions: the $N{=}300$ pilot (AUROC 0.563 [0.495, 0.631]) could not be distinguished from chance, and only the full run supported a verdict. Third, judge reliability is model- and domain-dependent (MCC 0.040–0.618 at a fixed threshold); thresholds must be calibrated per engagement, not baked into code. Fourth, missing evidence must be a first-class outcome; distinguishing "measured as bad" from "not measured" is what made the failures above visible.

These findings support the central claim: evidence completeness, judge validity, posterior regression risk, and fixed critical precedence must be first-class release criteria. Future work covers a paired Bayesian comparison, native evaluation of the remaining TRACe dimensions, and repeated-sampling analysis of per-case variance.

Use of Generative AI. Generative AI tools were used to assist with drafting and editing the manuscript and with implementing the evaluation harness and simulation code. All content, code, and reported numbers were reviewed and verified by the authors, who take full responsibility for them.

Ethical Considerations. The judge-layer study uses the public RAGBench benchmark. The statistical operating-characteristic study uses synthetic binary outcomes. No personal, client, or proprietary data are introduced by either gate-level experiment, and neither experiment calls an external model service. The deployed framework targets enterprise settings and includes local redaction of personally identifiable information before traces are persisted; gate outcomes are decision support with mandatory human review of non-promoted cases, not autonomous verdicts.

References

1. Bavaresco, A., Bernardi, R., Bertolazzi, L., Elliott, D., Fernández, R., Gatt, A., Ghaleb, E., Giulianelli, M., Hanna, M., Koller, A., et al.: LLMs instead of human judges? A large scale empirical study across 20 NLP evaluation tasks. arXiv preprint arXiv:2406.18403 (2024)
2. Chen, J., Lin, H., Han, X., Sun, L.: Benchmarking large language models in retrieval-augmented generation. In: Proceedings of the 38th AAAI Conference on Artificial Intelligence. pp. 17754–17762 (2024)
3. Chicco, D., Jurman, G.: The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics **21**(1), 6 (2020)

4. Cohen, J.: A coefficient of agreement for nominal scales. Educational and Psychological Measurement **20**(1), 37–46 (1960)
5. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman & Hall/CRC (1994)
6. Es, S., James, J., Espinosa-Anke, L., Schockaert, S.: RAGAs: Automated evaluation of retrieval augmented generation. In: Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations. pp. 150–158 (2024)
7. Friel, R., Belyi, M., Sanyal, A.: RAGBench: Explainable benchmark for retrieval-augmented generation systems. arXiv preprint arXiv:2407.11005 (2024)
8. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., Wang, H.: Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997 (2023)
9. Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., Rubin, D.B.: Bayesian Data Analysis. Chapman & Hall/CRC, 3rd edn. (2013)
10. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A., Fung, P.: Survey of hallucination in natural language generation. ACM Computing Surveys **55**(12), 1–38 (2023)
11. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. Biometrics **33**(1), 159–174 (1977)
12. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Advances in Neural Information Processing Systems 33 (NeurIPS). pp. 9459–9474 (2020)
13. Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., Zhu, C.: G-Eval: NLG evaluation using GPT-4 with better human alignment. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 2511–2522 (2023)
14. Rebedea, T., Dinu, R., Sreedhar, M.N., Parisien, C., Cohen, J.: NeMo Guardrails: A toolkit for controllable and safe LLM applications with programmable rails. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 431–445 (2023)
15. Saad-Falcon, J., Khattab, O., Potts, C., Zaharia, M.: ARES: An automated evaluation framework for retrieval-augmented generation systems. In: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 338–354 (2024)
16. Wang, P., Li, L., Chen, L., Cai, Z., Zhu, D., Lin, B., Cao, Y., Kong, L., Liu, Q., Liu, T., Sui, Z.: Large language models are not fair evaluators. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 9440–9450 (2024)
17. Zheng, L., Chiang, W.L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E.P., Zhang, H., Gonzalez, J.E., Stoica, I.: Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In: Advances in Neural Information Processing Systems 36 (NeurIPS), Datasets and Benchmarks Track (2023)