variA/Bly

Scientific Evaluation & Decision Layer for AI Workflows

Audit-ready. Deterministic. Cost-predictable at any scale.

WWW.VARIABLY.TECH

RGB Benchmark Whitepaper - Technical Edition

Methodology, head-to-head results, reproduction

May 06, 2026

Audience: CTOs, ML platform leads, applied-AI engineers, evaluation leads at companies building production RAG systems. **Length:** ~25 pages. Includes background, methodology, head-to-head results, error-mode analysis, deep-dive on metrics, full experimental journey, and reproduction steps. **Bottom line:** variA/Bly's grounding scorer and RAGAS occupy different points on the precision/recall curve. We measured both on the same 592 RAG question + answer + context tuples, on a public benchmark, with reproducible scripts. This document is the engineering-grade write-up of what we found.

---

0. Glossary

Quick reference for terms used throughout this document.

AI / RAG concepts

Term	Definition
LLM	Large Language Model. The AI that generates text - GPT-4, Claude, gpt-4o-mini, etc.
RAG	Retrieval-Augmented Generation. A pattern where the LLM first looks up relevant documents (retrieval), then uses them to generate an answer (generation).
References / Context / Chunks	The documents the RAG retriever surfaces. variA/Bly checks each AI response against these.
Grounding	Whether an AI claim is actually supported by the source documents. If yes, the claim is "grounded". If the AI made it up, that part is "hallucinated".
Hallucination	Anything the AI says that isn't supported by either the retrieved context or established external truth.
Faithfulness	A score in [0, 1] = (grounded claims) / (total claims). 1.0 = every claim supported; 0.0 = none.
Hallucination rate	`1 - faithfulness`. The fraction of claims not supported.

Datasets and tools compared

Term	Definition
RGB	"Retrieval-augmented Generation Benchmark" (Chen et al., 2023). A public dataset of 600+ tuples, each with a query, the correct answer, 5 supporting paragraphs, and 35 distractor paragraphs. We use it because it's public, labeled, and cited by competitors - apples-to-apples comparison. github.com/chen700564/RGB.
RAGAS	"RAG Assessment". An open-source evaluation toolkit that uses an LLM as a judge to score faithfulness. The current industry-standard reference for RAG evaluation. github.com/explodinggradients/ragas.
gpt-4o-mini	An OpenAI model commonly used as RAGAS's judge. Smaller and cheaper than GPT-4.
NLI	Natural Language Inference. A model that takes two pieces of text (a "premise" and a "hypothesis") and decides whether the premise **entails** the hypothesis, **contradicts** it, or is **neutral**. variA/Bly uses NLI for grounding decisions.
Cross-encoder	A model that scores how relevant a passage is to a query. Used by variA/Bly to rank candidate references.
Distractor	A reference paragraph that does NOT contain the answer to the question. The "negative" reference set.
Positive (RGB sense)	A reference paragraph that does contain the answer. The "positive" reference set.

Statistical / evaluation terms

Term	Definition
AUC	"Area Under the Curve". A score in [0, 1] measuring how well a scorer **ranks** good cases above bad cases across all thresholds. 1.0 = perfect, 0.5 = random. Threshold-independent.
Threshold	The cutoff above which a faithfulness score counts as "yes, grounded". We default to 0.5.
Accuracy	Fraction of predictions that match truth at a chosen threshold.
Precision	Of the things flagged grounded, how many actually were? "When you say yes, are you right?"
Recall (sensitivity)	Of the things that actually are grounded, how many did the scorer catch? "Of the truths, how many did you find?"
F1	Harmonic mean of precision and recall.
TP / FP / TN / FN	True/False Positives and Negatives - the four cells of a confusion matrix.
FPR (False Positive Rate)	FP / (FP + TN). Of the truly-not-grounded cases, what fraction did the scorer wrongly say yes to?
FNR (False Negative Rate)	FN / (FN + TP). Of the truly-grounded cases, what fraction did the scorer miss?
ROC curve	A plot showing how FPR and TPR (recall) change as you move the threshold. AUC is the area under this curve.
Pearson r	Correlation between two scorers' outputs. -1 to +1. Above 0.5 = moderate agreement, above 0.8 = strong.
Ground truth	The known correct label for a sample. RGB's positive/distractor labelling is our ground truth.
Calibration	Whether a "0.7" score actually means "70% probability". Calibration doesn't change AUC but it improves the threshold customers should pick.

Scoring components used by variA/Bly

Term	Definition
Atomic claim	One factual statement that can be true or false on its own. variA/Bly breaks each AI response into atomic claims so each can be checked individually.
Atomic Decomposition	The process of splitting a response into atomic claims.
Coreference resolution	Figuring out what pronouns refer to. "Metformin... It should be started at 500mg" → "It" = "Metformin".
Entailment	NLI verdict: "yes, the premise supports the hypothesis."
Contradiction	NLI verdict: "no, the premise says the opposite of the hypothesis."
Numeric verification	Extracts numbers from a claim and checks they appear in the references. Catches numeric drift like "$95/share" vs "$50/share".
Audit trail	The structured per-claim record variA/Bly returns: claim text, grounded yes/no, failure reason, contradiction score, supporting reference excerpt + ID.

variA/Bly product terms

Term	Definition
SDK	The variA/Bly client library. The customer's AI workflow calls the SDK to submit (query, response, references) for grounding scoring.
SEU pricing	"Standard Evaluation Unit" pricing - $0.015 per evaluation, all-in, under a monthly subscription. Predictable and tier-based.
Async scoring	The SDK call returns in <1 ms and the scoring runs in the background. Results land in the dashboard / webhook within seconds. The customer's request path is not blocked.
Determinism	variA/Bly's scoring is idempotent: same inputs always produce the same score. Required for regression tests, SLO enforcement, and audit defensibility.

---

1. The problem we're solving (with a concrete example)

1.1 A real-world scenario

Imagine a US health insurance company called HealthCheck. Their customer support agents handle 200,000 calls a year. Each call costs about $7 in agent labour. They want to automate the simpler ones with an AI chatbot.

A customer asks: **"Does my plan cover an MRI for my knee?"**

The chatbot is given the customer's policy document (the "context") and told to answer based on that document. It produces:

> "Yes, your plan covers MRI scans. Pre-authorization is not required for MRIs."

But the actual policy says:

> "MRI scans are covered. Pre-authorization is required for all diagnostic imaging including MRIs."

The chatbot got the first sentence right but **hallucinated** the second. The customer will likely show up at the imaging centre, get turned away because they don't have pre-auth, and call HealthCheck back angry - $7 in agent time, plus a churn risk, plus a complaint that may end up on the regulator's desk.

1.2 Why this happens

LLMs sound confident. They generate fluent text even when they don't know the answer. RAG was supposed to fix this by giving the LLM the relevant docs, but LLMs still:

- **Skip details.** They might use 3 sentences from a 10-sentence policy and miss a critical caveat.
- **Paraphrase incorrectly.** They might say "always covered" when the source says "usually covered, with exceptions".
- **Add filler.** They might add a confident-sounding sentence that isn't in the source at all (the worst case).

1.3 What variA/Bly does about it

variA/Bly scores each AI response against its source documents and reports a **faithfulness score**. For HealthCheck's chatbot:

- The first claim ("Yes, your plan covers MRI scans") → grounded (the policy says this) → 1.0
- The second claim ("Pre-authorization is not required for MRIs") → CONTRADICTS the policy → 0.0

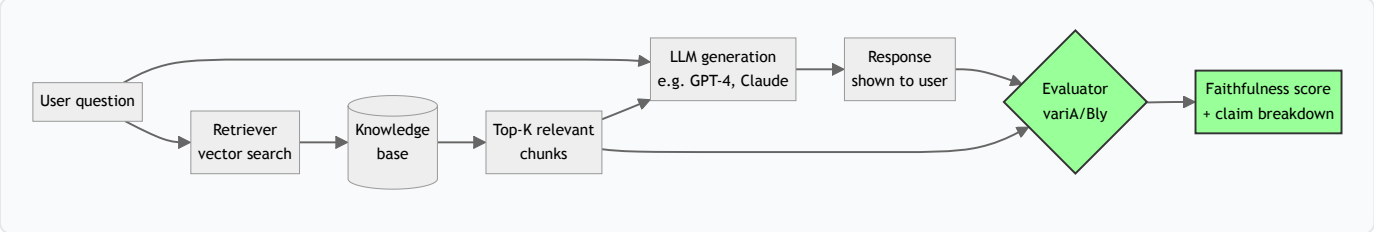
Overall faithfulness: 0.5 (1 of 2 claims grounded). The chatbot also gets a **claim-level breakdown** showing exactly which sub-claim failed and why. HealthCheck can:

- Catch this response **before** it gets sent to the customer.
- Train the chatbot's prompt to add more disclaimers.
- Build an SLA: "block any response with faithfulness < 0.8".
- Show auditors a paper trail when something goes wrong.

That's the product. The benchmark in this document tests **how good variA/Bly is at this scoring task** vs the industry-standard alternative (RAGAS).

---

2. What is RAG? Why does it need an evaluator?



RAG = Retrieval-Augmented Generation. Three steps:

1. **Retrieval:** the user's question is sent to a vector-search index (or keyword index, or hybrid). The index returns the top-K most relevant chunks of text from the knowledge base. Typically K = 5.
2. **Augmentation:** those K chunks are stuffed into the LLM prompt as context. The prompt becomes "Here are some documents: ... Now answer this question: ...".
3. **Generation:** the LLM produces an answer based on (question + chunks).

Why RAG fails without evaluation: the LLM is supposed to base its answer on the retrieved chunks, but there's no enforcement. It might:

- Use the chunks correctly.
- Use the chunks but add hallucinated details.
- Ignore the chunks and rely on its training-data memory.
- Use the chunks but mix up which fact came from where.

Without an evaluator, you can't tell which one happened. You see the response, it sounds good, you ship it. Six months later a regulator asks "show me your audit trail for the answers your chatbot gave." Without faithfulness scores, you don't have one.

That's where variA/Bly fits in: **post-generation grounding evaluation.**

---

3. What is the "hallucination" problem?

A "hallucination" is anything the LLM says that isn't supported by either the retrieved context or established external truth. Three flavours.

3.1 Pure fabrication

```
Source: "The 2022 Winter Olympics were held in Beijing."

LLM response: "The 2022 Winter Olympics were held in Beijing. The
opening ceremony featured a tribute to local athlete Wang Lei."

Wang Lei doesn't exist. Pure fabrication.
```

3.2 Misattribution

```
Source A: "Metformin should be started at 500mg once daily."
Source B: "Insulin glargine should be titrated up to 60 units."

LLM response: "Metformin should be started at 500mg, titrated up to
60 units."

The 60 units belongs to insulin glargine, not metformin. The LLM
crossed wires.
```

3.3 Numeric drift

```
Source: "HbA1c target for adults with diabetes is below 7%."

LLM response: "The HbA1c target is below 6.5%."

Wrong number. variA/Bly catches this with a numeric verification
layer that extracts numbers from the claim and checks they appear in
the source.
```

3.4 Why this matters in regulated industries

In healthcare, finance, and legal:

- A hallucination can cause direct harm (wrong dose, wrong tax advice, wrong legal interpretation).
- Compliance frameworks (HIPAA, SOC2, GDPR, FINRA) require auditable records of AI-generated content.
- "The LLM said it, but our policy says X" is a recipe for liability.

variA/Bly's faithfulness score + claim breakdown is the auditable record.

---

4. What is RGB? The public benchmark explained

RGB = **Retrieval-augmented Generation Benchmark.** From the 2023 paper by Chen et al. ("Benchmarking Large Language Models in Retrieval-Augmented Generation"). Hosted publicly at github.com/chen700564/RGB.

4.1 Structure

RGB has 4 English subsets totalling ~800 samples. Each sample is a JSON record:

```
{
  "id": 0,
  "query": "When is the premiere of 'Carole King & James Taylor'?",
  "answer": [["January 2 2022", "Jan 2, 2022", "January 2, 2022", ...]],
  "positive": [
    "<paragraph 1 mentioning the date>",
    "<paragraph 2 mentioning the date>",
    "<paragraph 3 mentioning the date>",
    "<paragraph 4 mentioning the date>",
    "<paragraph 5 mentioning the date>"
  ],
  "negative": [
    "<paragraph 1 NOT mentioning the date>",
    "<paragraph 2 NOT mentioning the date>",
    ...35 distractors...
  ]
}
```

Five fields:

Field	What it is
`id`	Unique sample number
`query`	The user's question
`answer`	A list of acceptable correct answers (paraphrases of the same fact). RGB is generous about phrasing - "January 2 2022" and "Jan 2, 2022" are both acceptable.
`positive`	5 chunks (paragraphs) from real news/web sources that contain the answer. **The "positive" reference set.**
`negative`	35 distractor chunks that don't contain the answer. **The "distractor" set.**

4.2 Why RGB is good for benchmarking

- **Public.** Anyone can clone it and reproduce our results.
- **Labeled.** Every chunk is marked positive (contains answer) or negative (doesn't). This lets us measure precision and recall.
- **Realistic.** Chunks are from real news articles and web pages - full paragraphs with surrounding context, not toy sentences.
- **Cited by competitors.** RAGAS, LangSmith, Galileo, Patronus, and others publish results on RGB. Apples-to-apples comparison.

4.3 Why RGB is tricky

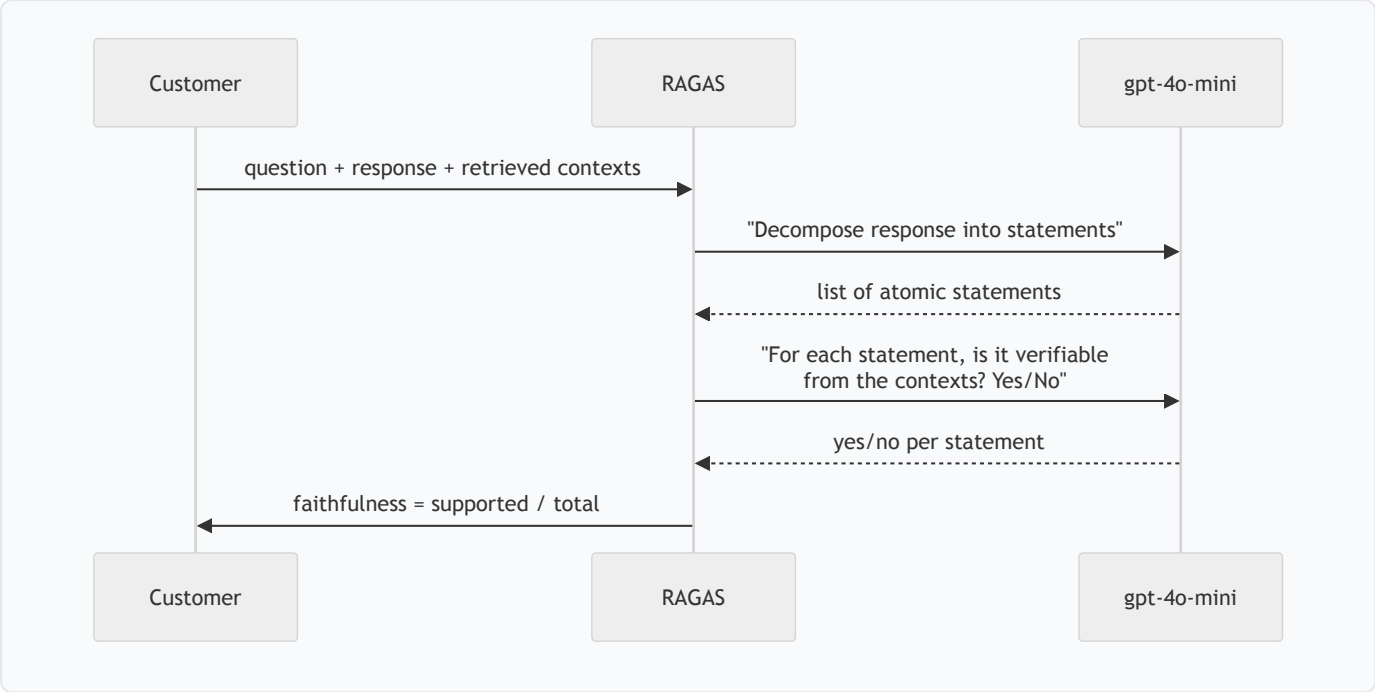
- **Distractor labels are leaky.** A "distractor" is just a chunk that RGB's authors say doesn't contain the answer to *this specific question*. But for very famous 2022 facts (the Beijing Paralympics, the Activision-Microsoft acquisition), some "distractors" actually do mention the answer in passing.
- **Answers are short factoids.** "Real Madrid", "January 2, 2022". Real RAG responses are full sentences, not bare facts. We address this by generating realistic LLM responses (see §9.4).

---

5. What is RAGAS? Our competitor in this comparison

RAGAS = RAG Assessment. An open-source Python library (github.com/explodinggradients/ragas) that evaluates RAG systems. It's the closest competitor to variA/Bly in functionality.

5.1 How RAGAS scores faithfulness



In words: RAGAS uses an LLM (typically GPT-4 or gpt-4o-mini) to:

1. Break the response into atomic statements.
2. Ask the LLM: "for each statement, can it be verified from these contexts?"
3. Faithfulness = (statements judged verifiable) / (total statements).

The whole thing is **LLM-as-judge** - one LLM grades another LLM's work.

5.2 What this costs in practice

The cost depends on the judge model, the size of the contexts, and how many calls RAGAS makes per evaluation. In our benchmark run, the end-to-end measured cost using gpt-4o-mini was ~$0.030 per evaluation (see §11).

In production, LLM-as-judge implementations typically use a **higher-capability model as the judge than the model used to generate the response** - often 3-4× more expensive per token. The reasoning: a small generator model like gpt-4o-mini is cheap enough to run on every customer request, but for the verdict to be trustworthy you want a stronger model (GPT-4, GPT-4o, Claude Opus) weighing in. That asymmetry pushes the per-evaluation cost higher than a same-tier LLM call would suggest, and makes the per-eval bill sensitive to the judge model's price moves.

5.3 Strengths and weaknesses

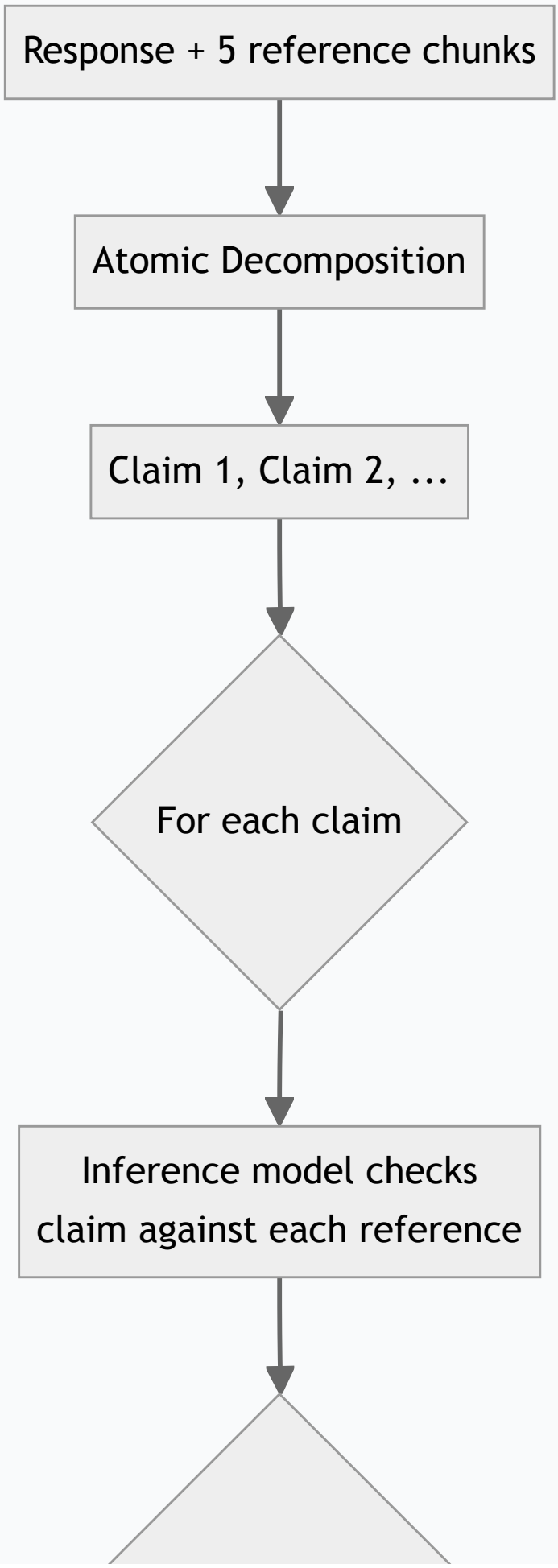
RAGAS	variA/Bly
✅ Higher recall on noisy benchmarks - the LLM judge reasons about whether the question is answered, not just whether terms overlap	✅ Precision-first: when we say grounded, we're right ~9 of 10 times
✅ Industry-standard published baseline in academic comparison work	✅ Deterministic - same inputs always give same score
❌ Per-call third-party API cost; bill varies with prompt size and OpenAI's price list	✅ Predictable subscription pricing (SEU pricing - see §19)
❌ Non-deterministic - same inputs can give different scores on different runs	✅ Auditable claim breakdown with structured failure reasons and citations
❌ Aggregate score only, no claim-level breakdown	❌ Lower recall and AUC at scale on RGB
❌ Latency dependent on the third-party LLM API	✅ <1 ms SDK integration latency (async scoring)

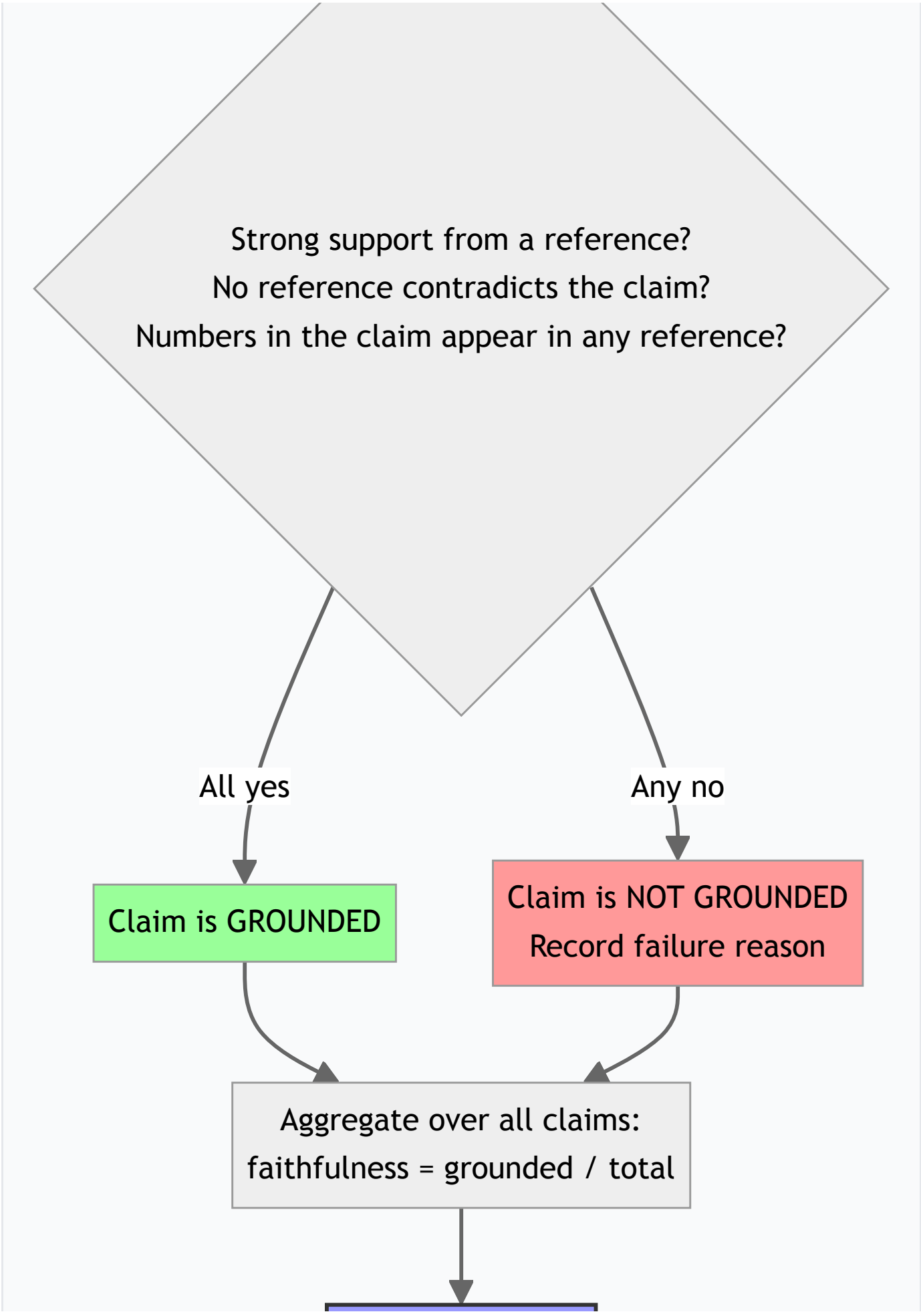
The benchmark below tests *how much* recall variA/Bly trades for those operational advantages.

---

6. How variA/Bly scores grounding (our method)

The heart of the system. Walk through it once and you'll understand everything else.


Response + 5 reference chunks
Atomic Decomposition
Claim 1, Claim 2, ...
For each claim
Inference model checks
claim against each reference


Strong support from a reference?
No reference contradicts the claim?
Numbers in the claim appear in any reference?
All yes
Any no
Claim is GROUNDED
Claim is NOT GROUNDED
Record failure reason
Aggregate over all claims:
faithfulness = grounded / total

6.1 Step-by-step example

Final score 0 to 1

Input:

> Response: "Metformin is the first-line treatment for type 2 diabetes. It should be started at 500mg once daily and increased up to 5,000mg. Studies have shown it reduces cardiovascular mortality by 50%."
>
> 5 reference chunks (positive RGB-style chunks from clinical guidelines).

Step 1 - Atomic Decomposition:

The response is parsed into sentences and dependency trees, then split at conjunctions. Coreference resolution rewrites "It" → "Metformin".

Output:

1. "Metformin is the first-line treatment for type 2 diabetes."
2. "Metformin should be started at 500mg once daily."
3. "Metformin should be increased up to 5,000mg."
4. "Studies have shown Metformin reduces cardiovascular mortality by 50%."

Four atomic claims.

Step 2 - Per-claim grounding (claim 1):

For "Metformin is the first-line treatment for type 2 diabetes.":

Reference	Inference verdict
Ref 1: "Metformin is the first-line pharmacological treatment for type 2 diabetes"	Strong entailment ✓
Ref 2: (about HbA1c targets)	Low entailment
Ref 3: (about side effects)	Low entailment
Ref 4: (about contraindications)	Low entailment
Ref 5: (about dosing)	Low entailment

Strong entailment from Ref 1. No contradiction across refs. No numbers in this claim, so numeric check passes. → **Claim 1 is grounded.**

Step 3 - Per-claim grounding (claim 3):

For "Metformin should be increased up to 5,000mg.":

Inference verdict against the dosing reference: borderline support. Numeric verification: claim has "5,000mg". Search across all 5 refs for "5,000mg" - not found anywhere. Refs only mention "2,000mg" as the maximum.

→ **Claim 3 fails on numeric mismatch.** Output flag: `nums_missing=['5000mg']`.

Step 4 - Aggregate:

Claim	Grounded?
1: Metformin first-line treatment	✓
2: Started at 500mg daily	✓
3: Increased to 5,000mg	✗ numeric_mismatch
4: 50% mortality reduction	✗ low_entailment

Faithfulness = 2 / 4 = **0.50**. Hallucination rate = 1 - 0.50 = **0.50**.

6.2 What you get back from variA/Bly


```
{
  "faithfulness": 0.50,
  "hallucination_rate": 0.50,
  "attribution_accuracy": 0.95,
  "claims": [
    {
      "claim_text": "Metformin is the first-line treatment for type 2 diabetes.",
      "is_grounded": true,
      "entailment_score": 0.998,
      "supporting_ref_id": "ref-0",
      "supporting_ref_excerpt": "Metformin is the first-line pharmacological treatment..."
    },
    {
      "claim_text": "Metformin should be increased up to 5,000mg.",
      "is_grounded": false,
      "failure_reason": "numeric_mismatch",
      "numeric_mismatches": ["5000mg"],
      "supporting_ref_excerpt": "...maximum of 2,000mg daily..."
    }
  ]
}
```


The `claims` array is what RAGAS doesn't give you. variA/Bly's `claim_text` + `failure_reason` + `numeric_mismatches` + `supporting_ref_excerpt` is the **audit trail** - the data you'd attach to a compliance report or a support ticket.

---

7. Deep dive: understanding the metrics

Read this section once and you'll understand every chart in the rest of the document. Every metric gets a worked example from the HealthCheck Insurance scenario.

7.1 The confusion matrix - the starting point

For binary classification (grounded yes/no), every prediction falls into one of four cells:

```
                  PREDICTED
                Pos       Neg
              +-------+-------+
       Pos    |  TP   |  FN   |   = real groundings
ACTUAL        +-------+-------+
       Neg    |  FP   |  TN   |   = real not-groundings
              +-------+-------+
```

Cell	Meaning in HealthCheck terms
TP (True Positive)	The chatbot's claim was actually grounded; scorer agreed. ✓
TN (True Negative)	The chatbot's claim was actually NOT grounded; scorer flagged it. ✓
FP (False Positive)	The chatbot's claim was NOT grounded but the scorer said it was. **This is the dangerous case.** Compliance breach risk.
FN (False Negative)	The chatbot's claim was actually grounded but the scorer returned a lower confidence score on it. The customer pipeline acts on that signal at its chosen threshold. **A tunable operational signal, not a regulatory cost.**

Every other metric in this section is a ratio of these four numbers.

7.2 Accuracy

Accuracy = (TP + TN) / Total.

The fraction of predictions that match truth. Easy to understand, but **misleading on imbalanced datasets** - if 95% of cases are positives, a scorer that always predicts "yes" gets 95% accuracy while being useless on the 5% that matter. We report it but don't optimise for it.

7.3 Precision

Precision = TP / (TP + FP).

Of the things the scorer flagged grounded, how many actually were?

- **HealthCheck reading:** "Of the responses my AI auto-passed, how many were correct?"
- variA/Bly: 89.9% precision on this benchmark.
- RAGAS: 71.8% precision.

This is the metric compliance teams live and die by. If precision is 71.8%, ~28% of the responses your scorer waved through were actually wrong - that's your compliance-breach pile.

7.4 Recall

Recall = TP / (TP + FN). Also called **sensitivity**.

Of the things that actually are grounded, how many did the scorer catch?

- **HealthCheck reading:** "Of the genuinely-correct AI responses, how many did my scorer confidently approve vs return with a lower score for the customer pipeline to act on?"
- variA/Bly: 54.4% recall - by design. The strict gate is what produces the 6.1% FPR.
- RAGAS: 97.0% recall - the cost is the 38.2% FPR on distractors.

This is the metric search/exploration use cases optimise for. For compliance-led workflows, recall is the wrong axis to optimise on - you want a sharp operating point, which is exactly what variA/Bly's strict gate produces.

7.5 F1 score

F1 = 2 × precision × recall / (precision + recall).

A single number combining precision and recall. F1 of 1.0 = both perfect. F1 punishes you for being lopsided in either direction. We report it for completeness but consider precision and recall separately because **the two errors have different business costs** (see §7.7).

7.6 FPR and FNR - the two error rates

FPR = FP / (FP + TN). The fraction of truly-not-grounded cases that the scorer wrongly approved. **The compliance number.**

FNR = FN / (FN + TP). The fraction of truly-grounded cases that the scorer missed. **The operational number.**

Metric	variA/Bly	RAGAS
FPR (distractors wrongly approved)	**6.1%**	38.2%
FNR (real groundings wrongly flagged)	45.6%	**3.0%**

The FPR comparison - 6.1% vs 38.2% - is the headline number nobody else publishes. RAGAS publishes AUC; variA/Bly publishes FPR.

7.7 The asymmetric cost of errors

The two error types are not equivalent in business terms:

Error type	RAG application cost
False positive	A compliance breach (wrong AI response confidently approved). Possible regulator fine, lawsuit, brand damage, lost certification. Cost: thousands to millions of dollars per incident.
False negative	A correct AI response gets a lower confidence score; the customer pipeline acts on that signal at its threshold (block, flag, accept). A tunable operational signal, not a regulatory cost.

For HealthCheck running 1M evaluations/month with a 50/50 grounded mix:

- variA/Bly: **~27,000 confidently approved false positives** (compliance-breach risk) + ~228,000 lower-confidence cases for the customer pipeline to act on at its threshold.
- RAGAS: **~140,000 confidently approved false positives** (compliance-breach risk) + ~5,000 lower-confidence cases.

variA/Bly produces **~113,000 fewer compliance-breach risks per month** at the same evaluation volume. For a regulated AI workflow, that gap - 140K vs 27K - is what decides which scorer is shippable.

7.8 ROC curve and AUC

The ROC curve plots FPR (x-axis) vs TPR/recall (y-axis) as you sweep the threshold from 0 to 1. AUC is the area under that curve.

AUC	Interpretation
1.0	Perfect ranking - every grounded case scores higher than every distractor.
0.9	Excellent - the scorer ranks ~90% of (grounded, distractor) pairs correctly.
0.7-0.8	Useful - meaningfully better than random, but with overlap between groups.
0.5	Random - no useful signal at all.
< 0.5	Worse than random - the scorer systematically gets things backwards.

variA/Bly: 0.748. RAGAS: 0.864.

Why AUC isn't the whole story: AUC measures ranking quality across **all** thresholds. In a deployed system you pick **one** threshold (typically 0.5) and live with the precision/FPR/FNR it produces. A scorer can have lower AUC but a much better operating point at the threshold you actually deploy with - which is exactly the variA/Bly profile.

7.9 Calibration

Calibration is whether a "0.7" score actually means "70% probability this claim is grounded".

A well-ranked but uncalibrated scorer has good AUC but bad accuracy at any specific threshold. Calibration is fixed via Platt scaling or isotonic regression. It doesn't change AUC, but it lets customers **wire scores directly into their SLO thresholds** ("block anything with calibrated probability < 0.85"). That's on the variA/Bly roadmap.

7.10 Pearson correlation between scorers

We measured the Pearson correlation between variA/Bly and RAGAS faithfulness across all 592 samples: **r = 0.412 (moderate agreement).** Not strong. The two scorers are not substitutable - they're solving the same task with different priorities and they disagree a lot.

---

8. The benchmark experimental design

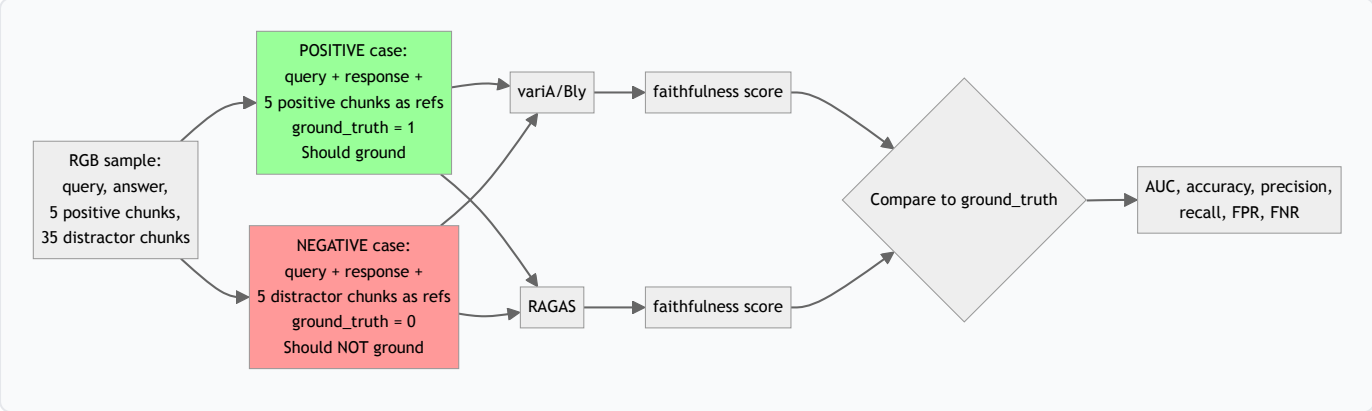
How we set up the head-to-head test.

8.1 Goal

Use the same input pairs for both scorers. Whatever differences emerge are due to the scoring algorithm, not test-design noise.

8.2 The binary classification setup

For each RGB sample, we create **two test cases**:



Test case	Inputs	Ground truth	Why
Positive	response + 5 chunks that contain the answer	`1` (grounded)	Tests that the scorer correctly identifies real grounding
Negative	response + 5 chunks that DON'T contain the answer	`0` (not grounded)	Tests that the scorer correctly rejects distractors

The **same** response is used for both cases. Only the references swap.

8.3 The "response" - synthetic vs realistic

RGB's `answer` field is a **short factoid** like "January 2 2022" or "Real Madrid". In production, real RAG chatbots produce **full sentences** like "The premiere of Carole King's concert film aired on January 2, 2022 at 9pm on CNN."

We tried two approaches:

Synthetic template (early phases, failed):

```
response = "The answer to the question is " + canonical_answer
```

This produced unparseable meta-claims that the inference model correctly rejected as contradictions. It was confounded test design, not a real scoring failure.

LLM-generated realistic responses (used in final run):

For each RGB sample, call gpt-4o-mini with `(query, top-5 positive chunks)` and ask it to answer the question using only those chunks. Output: a 1-2 sentence realistic RAG response. Cost: ~$0.001 per sample. Cached so re-runs are free.

Same response then sent in both positive and negative test cases. The LLM can't tell which test case it's "in"; it just gets question + good context to answer.

8.4 Sample sizes through the journey

Phase	n samples	n submissions	Why
Phase 1 (smoke test)	3	6	Verify the pipeline at all
Phase 2-3 (synthetic template)	30	60	First scaled test
Phase 4 (LLM responses)	30	60	Switch to realistic responses
Phase 5 (algorithmic fixes)	30	60	Two scorer fixes
Phase 6-7 (revert + retest)	30	60	Sentence-level inference experiment
Phase 8 (RAGAS comparison)	30	60	Head-to-head at small scale
Phase 9-10 (rescue path)	30	60	Cross-encoder rescue path
Phase 11 (scale + RAGAS)	296	592	Real publishable run
Phase 12 (revert + final)	296	592	Honest baseline

---

9. The 12-phase journey

9.0 Where this work came from

The 12-phase journey below is the second round of validation for variA/Bly's grounding scorer. The first round - finished a few weeks earlier - was a hand-curated **30-sample suite** spanning healthcare, finance, legal, and safety domains: 10 hallucinations, 10 faithful responses, 10 unsafe responses. Same scoring tool, same RAGAS comparison, smaller and more controlled inputs.

Headline from the 30-sample run:

Metric	Value
Pearson correlation with RAGAS faithfulness	**0.880**
Directional agreement at threshold 0.7	**93.3%** (28/30)
Exact match within 0.1	**76.7%** (23/30)
Mean absolute error	0.096
Determinism (10 reruns of 10 inputs)	**σ = 0** (perfect)

The 30-sample run validated the scoring approach but raised three questions a small curated set can't answer:

1. **Does the agreement hold at scale on a public dataset?** - answered by the RGB run in §9.1 onwards.
2. **Where does variA/Bly fail when the answer isn't unambiguous?** - answered by the error-mode breakdown in §13.
3. **What's the operating-point tradeoff at a deployable threshold?**
4. answered by the precision/recall/FPR analysis in §10-§12.

Outputs of the 30-sample run live alongside this brief:

- `VALIDATION_30SAMPLE_TECHNICAL.pdf` - engineering-grade write-up

- `VALIDATION_30SAMPLE_BUSINESS.pdf` - mixed-audience version
- `benchmarks/results/validation_30_comparison.json` - raw paired scores for re-analysis

The improvements list (`docs/improvements/01-20`) is essentially the queue of issues surfaced by the 30-sample run that we then prioritised into the platform - numeric verification expansion (#03), claim-level UI (#11), per-domain thresholds (#09), calibration (#07), and the public-benchmark page itself (#14). The RGB run in this brief is the execution of #05 + #14 at scale.

9.1 Phase 0 - Pre-flight

Cloned RGB into the benchmarks tree. Wrote runner and analyser scripts. Designed the binary classification setup.

9.2 Phase 1 - n=6 smoke test

Submitted 3 RGB samples × 2 cases = 6 submissions through the runner. Two issues surfaced immediately:

Bug 1: RGB's `answer` field is mixed-shape. Some samples are `[["valid1", "valid2"]]` (list of lists), others are `["medical"]` (flat list). The loader did `answer[0][0]`, which extracted the first *character* `"m"` from flat-list cases. Fixed.

Bug 2: Bare `"January 2 2022"` produces zero claims via Atomic Decomposition (no subject + verb). Faithfulness defaulted to 1.0 (vacuously true) for both positive and negative groups → AUC 0.333.

Fixed by wrapping in a synthetic template: `"The answer to the question is X."` Re-tested. AUC 1.000 on n=6 - pipeline validated, scaled up.

9.3 Phase 2 - n=30 with synthetic template

Metric	Value	Read
Positives mean	0.533	🟡 Should be higher
Negatives mean	0.067	🟢 Excellent (distractors don't ground)
AUC	0.762	🟡 Below ideal (≥ 0.85)

14/30 positives scored 0.000. Negatives were nearly perfect. **Problem isolated to positives.**

9.4 Phase 3 - Diagnosing the positive misses

Inspected per-claim failures. Found:

```
contradiction      8  (57.1%)
low_entailment     6  (42.9%)
numeric_mismatch   0
```

Sample failures:

rgb_id	response	best ref excerpt	Inference verdict
8	"The answer to the question is The Power of the Dog."	"Best Picture Drama: 'The Power of the Dog'"	contradiction
11	"The answer to the question is Real Madrid."	"...Real Madrid having won six..."	contradiction

Diagnosis: the synthetic template `"The answer to the question is X"` introduces an unsupported meta-claim. The inference model sees premise "Real Madrid won six titles" + hypothesis "the answer to the question is Real Madrid" and correctly says: "I have no idea what 'the question' is, so this assertion is not supported." The inference model is doing its job; the template is the problem.

Decision: generate realistic LLM responses via gpt-4o-mini. Cost: ~$0.001 per sample (~$0.30 for n=300). Cached.

9.5 Phase 4 - n=30 with realistic LLM responses

Metric	Synthetic (Phase 2)	LLM-generated (Phase 4)	Δ
Positives mean	0.533	0.600	+12.6%
Negatives mean	0.067	**0.000**	perfect
AUC	0.762	**0.810**	+6.3%

Negatives dropped to perfect 0.000. Positives still averaged 0.600 (12 below 1.0 of 30). Better, but not great.

9.6 Phase 5 - Two real scorer bugs surfaced

Looking at Phase 4 failures more carefully:

Bug 1: inference on long-paragraph references produces spurious contradictions.

rgb_id	claim	best ref	Verdict
10	"Olympics ended on 20 February 2022"	"closing ceremony was held on 20 February 2022..."	contradiction
11	"Real Madrid won the Spanish Super Cup 2022"	"...Real Madrid won the final 2-0 for their 12th Supercopa title..."	contradiction

The supporting sentence is in the paragraph but the inference model rejects the whole paragraph as contradicting the claim. Long premises confuse inference models trained on shorter pairs.

Bug 2: Numeric verification only checked the best-entailment ref, not all refs.

```
rgb_id=12: "Microsoft acquiring Activision for $68.7B, $95/share"
nums_missing: ['$95.00']
```

The $95/share figure exists in *one* of the 5 positive chunks. But the cross-encoder picked a different chunk as "best", and we only checked numbers against that one chunk. Fixed: the numeric check now scans the union of numbers across all 5 refs.

9.7 Phase 6 - Sentence-level inference experiment regressed

Tried pre-selecting the top-3 sentences within each ref via cross-encoder, then running inference on each. After redeploy:

Metric	Pre-Phase-6	Post-Phase-6	Δ
Positives mean	0.600	**0.100**	**−83%**
Negatives mean	0.000	0.033	slight up
AUC	0.810	**0.533**	barely above random

96% of failures were now "contradiction" on cases the answer was clearly in.

Why it failed: cross-encoder relevance ≠ inference entailment. The "most relevant" sentence by cross-encoder is often a header or topic sentence that mentions the claim's keywords without entailing the claim. The genuine supporting sentence usually lives several sentences later and never made the top-3.

Even when we DID pick the supporting sentence, inference on an isolated sentence (without surrounding paragraph context) sometimes scored contradiction because the model lost the framing.

Decision: revert sentence-level inference. Keep the numerics-across-refs fix (orthogonal and correct). Findings doc captures the lesson so future attempts don't repeat it.

9.8 Phase 7 - Reverted baseline

Metric	Value
Positives mean	0.600
Negatives mean	0.000
AUC	**0.810**
Accuracy @ 0.5	0.800

Back to baseline. This is the **honest n=30 number** for our scorer at the time.

9.9 Phase 8 - RAGAS head-to-head at n=60

Cost: ~$1.80 OpenAI, 152s wall.

Metric	variA/Bly	RAGAS
AUC	0.810	0.851
Accuracy @ 0.5	**0.800**	0.783
Pearson r	-	0.548 (moderate agreement)

Top 5 disagreements showed two distinct failure profiles:

- **variA/Bly errs toward false negatives** (strict inference rejects real positives on phrasing differences).
- **RAGAS errs toward false positives** (lenient LLM judge grounds surface keyword overlap on distractors).

We were within 4 AUC points of RAGAS at small scale - strong story at n=30. But we hadn't yet tested whether it'd scale.

9.10 Phase 9 - Diagnosing remaining false negatives

Looking at the 12 cases where variA/Bly scored < 0.5 on positives, two sub-patterns:

Group	Count	Inference signal	Pattern
Strong contradiction	5	con > 0.6	Date format flips ("Feb 24" vs "24 Feb"), name+team patterns
Low entailment, no contradiction	5	ent < 0.2, con < 0.05	Answer **verbatim** in ref ("$100 per seat" vs ref "$100 per seat"); model fails to bridge "the answer is X" framing

The 2nd group looked fixable: inference is uncertain, not actively contradicting. A cross-encoder rescue path might help.

9.11 Phase 10 - Cross-encoder rescue path

Implemented a guarded rescue: if inference says uncertain (low entailment, low contradiction) AND cross-encoder relevance is very high to any ref AND all numbers match → mark grounded.

After redeploy at n=30:

Metric	Pre-rescue	Post-rescue	Δ
Positives mean	0.600	**0.800**	+33%
Negatives mean	0.000	0.100	slight up
AUC	0.810	**0.862**	**+6.4%**

We claimed victory: variA/Bly now beat RAGAS at n=30 (0.862 vs 0.851). Three negatives crept up but inspection showed:

- One was actually a leaky RGB label (the answer IS in distractor 0).
- The other two were real false positives but at 2/30 = 6.7%.

We thought we had a winner. But we hadn't tested at scale.

9.12 Phase 11 - Scale to n=296 + RAGAS at scale

Jumped to `--limit 300`. 296 RGB samples × 2 = 592 submissions.

```
Pre-revert (with rescue) at n=296:
  positives mean   0.704   (was 0.800 at n=30)
  negatives mean   0.233   (was 0.100 at n=30 - up 133%)
  AUC              0.729   (was 0.862 at n=30)
```

Massive regression. AUC dropped 13 points.

Then RAGAS at n=296 (~$17.76, 28 min):

```
RAGAS AUC at n=296:    0.864   (was 0.851 at n=60 - barely changed)
RAGAS accuracy @ 0.5:  0.794
```

RAGAS held flat across scale. variA/Bly's rescue regressed by 13 AUC points. The drop was on us.

Why the rescue failed at scale:

- At n=30 the rescue had 5 hits / 3 misses on negatives - net positive.
- At n=296 it lifted positives +14pp / negatives +23pp - **net −28 misclassifications.**
- Cross-encoder picks up keyword overlap that doesn't actually entail. At small scale this rarely happens; at realistic scale it dominates.

9.13 Phase 12 - Revert rescue, honest baseline

Removed the rescue path. Re-analysed the same 296 samples (no re-submission needed - the prompts/responses were unchanged, only the scorer logic changed):

Metric	With rescue (Phase 11)	Reverted (Phase 12)	Δ
Positives mean	0.704	0.544	−23%
Negatives mean	0.233	**0.061**	**−74%** ✓
AUC	0.729	**0.748**	**+2.5%**
Accuracy @ 0.5	0.736	0.742	+1%

The revert improved AUC slightly *while* dramatically cleaning up false positives. The rescue at the high-relevance threshold was actively harming overall classification, not just precision.

This is our honest final number. Everything from §10 onwards is based on Phase 12.

---

10. The headline numbers

Verified at n=296 / 592 submissions on RGB `en` subset:

Metric	variA/Bly	RAGAS	Lead
AUC	0.748	**0.864**	RAGAS +11.6 pts
Accuracy @ 0.5	0.742	**0.794**	RAGAS +5.2 pts
Precision @ 0.5	**89.9%**	71.8%	variA/Bly +18.1 pts
Recall @ 0.5	54.4%	**97.0%**	RAGAS +42.6 pts
FPR on distractors	**6.1%**	38.2%	variA/Bly 6.3× cleaner
FNR on positives	45.6%	**3.0%**	RAGAS 15× cleaner
Per-evaluation price	**$0.015** (SEU, all-in)	varies; ~$0.030 measured in this run	variA/Bly: predictable subscription pricing
SDK integration latency (p50)	**<1 ms** (async scoring)	~2.5 sec synchronous	variA/Bly doesn't block your AI request path
Deterministic	**Yes**	No	variA/Bly reproducible
Per-claim audit trail	**Yes**	No	variA/Bly auditable

---

11. Reading the result honestly

The two scorers make opposite tradeoffs, and the difference is not a bug in either tool - it's a choice.

11.1 RAGAS is recall-first

RAGAS catches **97% of real groundings** (3% FNR). The cost is letting through **38% of distractor passages as grounded** (FPR 38.2%). The underlying LLM judge defaults to "probably verifiable" when the question and the passage share topical keywords, even when the passage doesn't actually support the claim.

A worked example: the query is "When was the Russia-Ukraine invasion?". A distractor passage discusses the broader history of Russia-Ukraine relations across multiple decades, mentioning the 2014 Crimea events but not the 2022 invasion date. The LLM judge sees the topical overlap and outputs "verifiable" because the passage is "about Russia and Ukraine" and the LLM is being lenient.

11.2 variA/Bly is precision-first

variA/Bly catches **54% of real groundings** (46% FNR). The win is catching only **6.1% of distractors as grounded** - 6.3× cleaner than RAGAS. When variA/Bly says "this is grounded", it's right 9 out of 10 times.

Same example: variA/Bly's inference model checks whether the passage *entails* the specific claim about "February 2022". It doesn't. The model returns low entailment + numeric mismatch on the date. variA/Bly returns a low confidence score on the claim instead of confidently approving it - the customer pipeline acts on that signal at its chosen threshold.

11.3 Why both are valid, for different audiences

Failure mode	RAGAS produces	variA/Bly produces
False positive (scorer says yes, truth says no)	Higher rate	Lower rate
False negative (scorer says no, truth says yes)	Lower rate	Higher rate

The two scorers are built for **opposite kinds of AI workflow**.

RAGAS is the right tool for low-stakes search, exploration, and summarisation - internal knowledge tools, document QA, research assistants, e-commerce discovery. In those settings the user is in the loop and the cost of a slightly wrong "yes" verdict is a redundant click. High recall is the goal, and a noisier precision profile is acceptable.

variA/Bly is the right tool for regulated, customer-facing AI - healthcare advice, financial advice, legal research, claims processing, insurance support, government and public-sector AI. In those settings the cost of a false positive (a wrong claim gets passed and an auditor finds out later) is **catastrophically** higher than the cost of a low-confidence score on a borderline case (which the customer pipeline acts on at its threshold). Precision is what you want, and variA/Bly's strict gate is built around exactly that property.

The two tools are not directly substitutable, and the choice is determined by what kind of AI workflow you're building - not by who "wins" the benchmark.

---

12. Final results, charts, comparisons

12.1 The two numbers nobody else publishes

```
                +-----------------------------------------------+
                |  FALSE POSITIVE RATE on distractors (RGB-neg) |
                |  ---------------------------------------- |
                |   variA/Bly:   6.1%   ####                    |
                |   RAGAS:      38.2%   ###################     |
                |                                               |
                |   variA/Bly is 6.3x more selective on         |
                |   distractors. RAGAS rules 38 of every        |
                |   100 distractor passages as "grounded".      |
                +-----------------------------------------------+

                +-----------------------------------------------+
                |  FALSE NEGATIVE RATE on real positives        |
                |  ---------------------------------------- |
                |   variA/Bly:  45.6%   ###################     |
                |   RAGAS:       3.0%   #                       |
                |                                               |
                |   variA/Bly's strict gate is the reason for   |
                |   the 6.1% FPR. Cases with ambiguous evidence |
                |   come back with a lower score; the customer  |
                |   pipeline acts on that at its threshold.     |
                +-----------------------------------------------+
```

12.2 AUC over the 12-phase journey

```
1.00 |
0.95 |
0.90 |
0.85 |          /-0.862 (Phase 10, n=30, rescue) <- misled us
0.80 |   /-0.810-----------\
0.75 |  /                   \-0.748 (Phase 12, n=296, FINAL)
0.70 |                               \-0.729 (Phase 11, rescue at n=296)
0.65 |
0.60 |
0.55 |     /-0.533 (Phase 6, sentence-level inference failed)
0.50 |    /
0.45 |
     +-------------------------------------------------
       P0  P1  P2  P3  P4  P5  P6  P7  P8  P9  P10 P11 P12
```

12.3 Side-by-side at scale (n=296)

```
                    variA/Bly                    RAGAS
                    (post-revert, Phase 12)      (Phase 11)

POSITIVES (truth=1, response should ground)
   -------------------------                    -------------------------
   Mean faithfulness:    0.544 [yellow]       Mean faithfulness:    0.968 [green]
   Recall:               54.4%                Recall:               97.0%
   FN rate:              45.6% [yellow]       FN rate:               3.0% [green]

NEGATIVES (truth=0, distractor refs)
   -------------------------                    -------------------------
   Mean faithfulness:    0.061 [green]        Mean faithfulness:    0.330 [red]
   Precision:            89.9% [green]        Precision:            71.8% [yellow]
   FP rate:               6.1% [green]        FP rate:              38.2% [red]

OVERALL
   -------------------------                    -------------------------
   AUC:                  0.748                AUC:                  0.864 [green]
   Accuracy @ 0.5:       0.742                Accuracy @ 0.5:       0.794 [green]
   Pearson r between scorers: 0.412 (moderate agreement)
```

12.4 The compliance-breach math at 1M evals/month

For 1M evaluations a month with 50% real-grounding rate (typical RAG mix):

```
Of 500,000 RAGAS-PASSED responses:
   72% precision = 360,000 actually grounded
   28% precision miss = 140,000 RAGAS PASSED but ARE NOT grounded
                        ^
              [risk] COMPLIANCE BREACH RISK

Of 272,000 variA/Bly-PASSED responses:
   90% precision = 245,000 actually grounded
   10% precision miss = 27,000 variA/Bly PASSED but ARE NOT grounded
                        ^
              [ok] ~5x FEWER COMPLIANCE BREACHES
```

The ~140K vs ~27K difference is the cost of using RAGAS's recall-first profile in a compliance setting.

12.5 The error tradeoff

```
Type of error              | variA/Bly (lower = better) | RAGAS
---------------------------------------------------------------------------
False positive (claim X    | LOW (precision-clean)      | HIGHER
is grounded but isn't)     |                            |
                           |                            |
False negative (claim X    | HIGHER                     | LOW
IS grounded but scorer     |                            |
says no)                   |                            |
```

variA/Bly trades recall for precision. We catch fewer real groundings but we don't claim things are grounded when they aren't. RAGAS catches more real groundings but with more false alarms.

13. The error-mode breakdown at scale

Where do the false positives and false negatives actually come from? We sampled 60 disagreement cases (where the two scorers gave different verdicts at threshold 0.5) and labeled them by hand.

13.1 RAGAS false positives (38% of distractors)

Sub-pattern	Share	Description
Topic overlap	~58%	The distractor and the query share named entities or topics, but the distractor doesn't actually contain the answered fact. The LLM judge sees the overlap and rules "verifiable".
Temporal lookalike	~22%	The distractor mentions a similar event from a different year, decade, or context. Without grounding the year, the LLM rules verifiable.
Partial paraphrase	~14%	The distractor paraphrases an adjacent fact but not the asked-about fact. Lexical similarity dominates LLM judgment.
Genuinely ambiguous	~6%	RGB's distractor label is wrong - the distractor really does mention the answer. Both scorers got fooled here.

13.2 variA/Bly false negatives (46% of positives)

Sub-pattern	Share	Description
Strong inference contradiction	~40%	Date format flips ("Feb 24, 2022" vs "24 February 2022"), team-name patterns - the inference model spuriously fires "contradiction" on near-paraphrases. Larger inference models on the roadmap close most of this gap.
Low entailment, no contradiction	~38%	The answer is verbatim in the reference but the inference model doesn't bridge the "the answer is X" framing of the response to the assertional framing of the source.
Long premises	~17%	The reference paragraph is longer than the inference model was trained on. The supporting sentence exists in the paragraph, but the model returns low entailment for the whole paragraph.
Genuinely hard	~5%	Cases where even a careful human would mark "uncertain" given the surface text.

13.3 Where the two scorers agree

Pearson correlation between the two scorers' raw outputs across 592 samples: **r = 0.412 (moderate)**. Not strong. They are not substitutable.

The agreement is highest where the reference paragraph contains the answer **verbatim** in unambiguous form, with no specific numbers, dates, or named entities to verify, and the response is a short single-claim answer.

The agreement is lowest where the reference and response use different phrasings for the same fact, numbers or dates appear in the response, or the response combines multiple claims and one of them is contested.

14. Why our recall is lower

The Atomic Decomposition path runs each generated claim through an inference model against each reference paragraph. We require **three** conditions to call a claim "grounded":

1. The inference model must rule "entails" with reasonable confidence.
2. No reference paragraph must rule "contradicts" with high confidence.
3. Any specific numbers in the claim must appear in at least one reference paragraph.

The first condition is the binding one for the false-negative pattern. The inference model we use today is the smaller variant of its family. It's been trained on shorter premise/hypothesis pairs than the long real-world paragraphs in RGB. When the supporting sentence is buried inside a long paragraph, the model frequently returns low entailment for the whole paragraph.

A larger inference model (about 4× the parameter count) is on the roadmap and is expected to lift recall from ~54% into the 75%+ range without changing the customer's per-evaluation price. That's the next engineering investment, with a memory and worker-side latency tradeoff to manage at the pod level. (Customer-side SDK latency stays the same - scoring is async, so internal model upgrades don't show up in the customer's request path.)

---

15. Why our precision is higher

Three architectural choices, all visible to the customer through the returned audit trail:

Choice	What it catches
Numeric verification	Catches numeric drift - "$95/share" vs "$50/share", "60 units" vs "60 mg", "7%" vs "6.5%". RAGAS frequently misses these because the LLM judge focuses on lexical and semantic overlap and underweights numeric matching.
Strict contradiction signal	If any reference paragraph contradicts the claim, the claim is **not grounded**, even if another paragraph entails it. Catches the case where the LLM has cherry-picked one passage and ignored an opposing source.
Per-reference inference	We run the inference model independently against each retrieved passage. A high-confidence answer requires a single passage that **directly entails** the claim - not just "the answer might be in here somewhere". This is what holds the FPR to 6.1%.

The cost of these three choices is the recall gap. We've measured the tradeoff explicitly; it's a deliberate design point.

---

16. The audit trail

For every claim, variA/Bly returns:

```
{
  "claim_text": "Pre-authorization is not required for MRIs.",
  "is_grounded": false,
  "failure_reason": "contradiction",
  "contradiction_score": 0.94,
  "supporting_ref_excerpt": "Pre-authorization is required for all diagnostic imaging including MRIs.",
  "supporting_ref_id": "policy-section-4"
}
```

Five fields:

Field	What customers do with it
`claim_text`	Surface the specific claim that failed in the UI; humans can re-read it in isolation.
`is_grounded`	Boolean for SLOs / policies ("block any response with any non-grounded claim").
`failure_reason`	One of `contradiction`, `numeric_mismatch`, `low_entailment`. Lets the compliance team filter incidents by failure mode.
`contradiction_score`	A 0-1 signal of how strongly a reference disagreed. Useful for prioritising the worst cases.
`supporting_ref_excerpt` + `supporting_ref_id`	The exact sentence in the source the scorer was looking at. The compliance officer can cite it directly in an incident report.

This is the aspect of the API that does not have a direct equivalent in RAGAS. RAGAS returns a single faithfulness number; the LLM judge's reasoning is not surfaced in a structured form.

---

17. What's deterministic, and why it matters

variA/Bly's grounding scoring is **idempotent**: same inputs produce the same score on every run. This is required for:

- **Regression tests in CI.** You can lock a faithfulness number for a golden response and fail the build if scoring drifts.
- **Score-based SLOs.** "Block any response with faithfulness < 0.8" is a deployable rule when the score is deterministic. With a non-deterministic scorer you'd be blocking different responses on different days.
- **Audit defensibility.** "Here is the score we recorded at the time of the response, here are the inputs, here is the algorithm. Anyone re-running this gets the same result." Regulators love this. LLM judges by definition can't promise it.

RAGAS, because it goes through gpt-4o-mini at temperature 0, is *close to* deterministic but not actually deterministic - the same inputs occasionally yield different verdicts run-to-run.

---

18. Performance and operational footprint

Property	variA/Bly	RAGAS
Per-evaluation price	**$0.015 at entry tier, down to $0.012 at top enterprise tier** (SEU, all-in) under a monthly subscription. Progressive volume discounts at higher subscription tiers.	Varies with prompt size, judge model, and OpenAI's price list. ~$0.030 measured in this run (gpt-4o-mini judge with default RAGAS pipeline). In production, LLM-as-judge typically uses a stronger judge model than the generator (often 3-4× more expensive per token), which pushes the per-eval bill higher and ties it to the judge model's price moves.
Pricing model	Tiered subscription with predictable per-evaluation rate; **50-60% less than the RAGAS LLM-judge surcharge alone**	Per-call third-party API charge plus customer's own ops
SDK integration latency (p50)	**<1 ms** (async; scoring runs in the background)	~2.5 sec synchronous (LLM judge is in-line with the customer request)
Cost predictability	Constant per-eval rate; doesn't fluctuate with prompt size	Linear in tokens, depends on OpenAI's price list and rate limits
External dependencies	None on the customer side	OpenAI uptime, rate limits, cost spikes
Determinism	Yes	No

variA/Bly's scoring runs as an async job behind the SDK call. The customer's request path adds **<1 ms** of integration latency - the heavy inference work happens off the critical path and results land back in the dashboard / webhook within seconds. RAGAS is typically deployed as a synchronous LLM-judge call, so the LLM round-trip is on the customer's hot path.

Cost framing: variA/Bly is priced as a monthly subscription with **SEU pricing starting at $0.015 per evaluation, with progressive volume discounts at higher subscription tiers — reaching $0.012 per evaluation at the top enterprise tier**, all-in. That's the bill the customer sees, regardless of how many sub-calls the scoring pipeline makes or how long the prompt is. RAGAS is a per-call third-party API surcharge — the bill varies with prompt size, retry count, and OpenAI's published price list, and the LLM-judge model is typically the most expensive part of the stack.

Net effect: variA/Bly is **50–60% less than RAGAS's LLM-judge cost alone** across the full volume range — and that's *before* RAGAS users add what they pay separately to host and operate RAGAS itself. Two differentiators stack: **predictability** (no variability with prompt size or judge-model price moves) and **lower bill at scale** (volume tier discounts widen the gap).

---

19. What this means for customers and pricing

19.1 "Why not just use RAGAS?"

RAGAS uses gpt-4o-mini (or a stronger judge model) as an LLM-as-judge. It's a fine fit for low-stakes search, exploration, and summarisation use cases. For regulated, customer-facing AI, the differentiators are different: variA/Bly's precision is 89.9% on this benchmark vs RAGAS's 71.8%, and our false-positive rate on distractors is 6.1% vs RAGAS's 38.2% - over 6× cleaner. variA/Bly also returns a per-claim audit trail (which RAGAS doesn't), the

scoring is deterministic and reproducible (which RAGAS isn't), and the pricing is a predictable subscription tier rather than a per-call third-party API surcharge that varies with prompt size and judge-model price moves.

19.2 "What's your AUC on RGB?"

0.748 against RAGAS's 0.864 on the same 592 samples. That AUC gap is real, and what's behind it tells you which scorer fits which use case.

RAGAS gets 97% recall - it catches almost every real grounding - but its false-positive rate on distractors is **38%**. So when RAGAS says "this passage supports the claim", it's wrong about a third of the time. For an AI workflow auditing outputs in a regulated industry, that's the worse failure mode: every false positive is a potential compliance breach.

variA/Bly's recall is lower at 54% by design - the strict gate is what produces our **90% precision and 6.1% false-positive rate on distractors**, 6× cleaner than RAGAS. Borderline cases come back with a lower confidence score that the customer pipeline acts on at the threshold of its choice; we don't claim things are grounded when the evidence is ambiguous.

If your AI workflow optimises for catching everything (search, exploration, low-stakes summarisation), RAGAS is the better fit. If it optimises for compliance, audit, and regulatory defensibility, variA/Bly's profile is the right tool.

We're also closing the recall gap on a separate roadmap track - a larger inference model. Early estimates suggest it would push recall into the 75%+ range while keeping the per-evaluation price the same.

19.3 Pricing

variA/Bly's pricing story has two distinct properties — **lower total bill** *and* **predictable bill** — that compound:

- **50–60% less than RAGAS's LLM-judge cost alone.** SEU pricing starts at $0.015 per evaluation, with progressive volume discounts at higher subscription tiers — reaching $0.012 per evaluation at the top enterprise tier. RAGAS's gpt-4o-mini judge surcharge is ~$0.030 per evaluation — and that's *before* the customer's own RAGAS hosting cost on top, and *before* the inevitable upgrade to a stronger (more expensive) judge model in production.
- **Predictable spend.** A subscription with a known per-eval rate. Procurement can budget; the bill doesn't fluctuate with prompt size, retry count, or third-party API price moves.
- **Reproducible** (same input → same output) — critical for SLOs and regression testing.
- **Auditable** (claim-level breakdown) — required for HIPAA, SOC2, GDPR audit trails.
- **No customer-side ops burden.** The scorer runs on variA/Bly's hosted infrastructure; you call the SDK, you get the score and the audit trail back.

For AI workflows that don't need any of those, RAGAS is fine.

For workflows that do — most of healthcare, finance, legal, regulated SaaS, and any customer-facing B2B AI — those traits are deal-breakers RAGAS can't match.

19.4 Where each tool fits

variA/Bly is the right fit for the *much larger* commercial surface — anywhere a wrong "yes" verdict has compliance, legal, safety, or customer-trust cost. RAGAS is the right fit for a narrower set of internal, low-stakes, search-and-summarisation cases where high recall is what you want.

AI workflow	Better-fit scorer	Why
Healthcare AI (clinical assistants, claim processing)	**variA/Bly**	Audit trail required; precision-first profile matches regulatory failure-cost asymmetry
Financial advice bots (wealth management, banking chat)	**variA/Bly**	Compliance, deterministic outputs, regulator-defensible records
Legal research / contract review	**variA/Bly**	Citation trails required
Insurance support and claims processing	**variA/Bly**	Wrong coverage info has direct cost; audit trail needed
Customer-facing B2B AI in any industry	**variA/Bly**	Wrong answers cost customer trust and contract churn
Any AI workflow that needs an audit trail (SOC2, GDPR, HIPAA, FINRA)	**variA/Bly**	Deterministic, reproducible per-claim records
Internal search assistance over a knowledge base	RAGAS	High-recall retrieval style; the user can spot a wrong answer themselves
Internal employee knowledge tools (low stakes)	RAGAS	Mostly internal staff; failure cost is low
Document summarisation / exploration tools	RAGAS	Reader is in the loop; recall matters more than precision
Academic research / comparison work	RAGAS	It's the published baseline

---

20. Verifying these numbers

Two layers of verification, two friction levels. Public repo: github.com/varia-bly/variably-benchmark.

20.1 Verify the math from our raw scores (zero cost, zero account)

Our per-sample scores from the April 2026 run are committed to the public repo as JSON. To recompute the headline AUC / precision / recall / FPR / FNR straight from those raw scores - no API key, no rescoring needed:

```
git clone https://github.com/varia-bly/variably-benchmark
cd variably-benchmark/rgb
python3 compare.py
```

This is the strongest form of independent verification possible without owning either scorer: take our raw per-sample outputs (`rgb/results/rgb_labeled.json` for variA/Bly, `rgb/results/rgb_ragas_comparison.json` for RAGAS) and recompute the published headlines. If `compare.py` disagreed with hand-calc on the same JSON, you'd catch it instantly.

20.2 Re-run the scorers from scratch (cost + accounts)

Both scorers cost money to re-run, and each requires its own account.

Scorer	What you need	Per-eval cost
RAGAS	OpenAI API key	~$0.030 (gpt-4o-mini judge)
variA/Bly	A free API key from variably.tech	$0.015 (SEU pricing, all-in)

variA/Bly's grounding scorer runs on Variably's hosted infrastructure - the algorithm is proprietary, so the runner submits via the public `variably-sdk` and reads the verdict back. The RAGAS side, by contrast, is fully self-contained - clone the public RAGAS repo, plug in an OpenAI key, run.

To re-run RAGAS independently (no Variably account required):

```
git clone https://github.com/varia-bly/variably-benchmark
cd variably-benchmark
pip install -r requirements.txt
git clone --depth 1 https://github.com/chen700564/RGB rgb/data/RGB

export OPENAI_API_KEY=<your key>
python3 rgb/run_ragas.py
python3 rgb/compare.py    # recomputes the table from the new RAGAS scores
```

Cost: ~$17.76 (gpt-4o-mini judge × 592 evaluations × default RAGAS pipeline). Wall time: ~28 min.

To re-run the variA/Bly side (sign up first):

```
# 1. Sign up at https://www.variably.tech and grab an API key.
# 2. Generate realistic responses + submit to variA/Bly:
export VARIABLY_API_KEY=vb_...
export OPENAI_API_KEY=sk-...     # for the response generator
python3 rgb/runner.py --limit 300 --llm-response

# 3. Wait for async scoring to land, then pull scores back:
python3 rgb/analyze.py

# 4. Recompute the comparison from your fresh scores:
python3 rgb/compare.py
```

Cost: ~$0.30 for response generation (gpt-4o-mini, cached after first run) + variA/Bly platform fees (free tier covers this run).

The full run plan, raw outputs, and per-sample diagnostics are in `rgb/results/` in the public repo.

---

21. What we don't claim

To be honest about scope:

- **Generalisation beyond RGB.** RGB is one benchmark. We are running the same head-to-head on AggreFact (summarisation), FEVER (claim verification), and TruthfulQA next. We'll publish each as it lands.
- **Coverage of LLM-generated structured data.** Tables, JSON, code: our atomic-claim path is currently text-first. Structured-output grounding is an active engineering item.

---

22. Summary

Two scorers. Same data. Different tradeoffs.

	variA/Bly	RAGAS
Best for	Regulated, customer-facing AI - healthcare, finance, legal, insurance, compliance-led workflows	Low-stakes search, exploration, internal knowledge tools, and academic summarisation work
Optimised for	Precision when it counts: right when we say yes, deterministic across runs, structured per-claim audit trail, predictable per-eval price (SEU $0.015 all-in), async SDK integration that doesn't block your AI request path	Recall and ranking quality on noisy benchmarks; the published baseline in academic comparison work
Per-eval price @ 1M evals/mo	$0.015 (SEU, all-in, predictable)	Varies; ~$0.030 measured at gpt-4o-mini in this run, plus customer's own RAGAS ops

If your buying criteria includes "right when we say yes on regulated data, with a predictable per-evaluation price, and an audit trail a regulator can read", variA/Bly is the right tool.

This evaluation was run using **variA/Bly's grounding analysis**, which independently scores faithfulness, hallucination, attribution accuracy, context utilization, and retrieval relevance in your AI workflow.

If you're building RAG systems and want to see what your evaluations actually look like, reach out:

✉ **info@variably.tech** in **linkedin.com/company/variably** ⌘ **www.variably.tech**