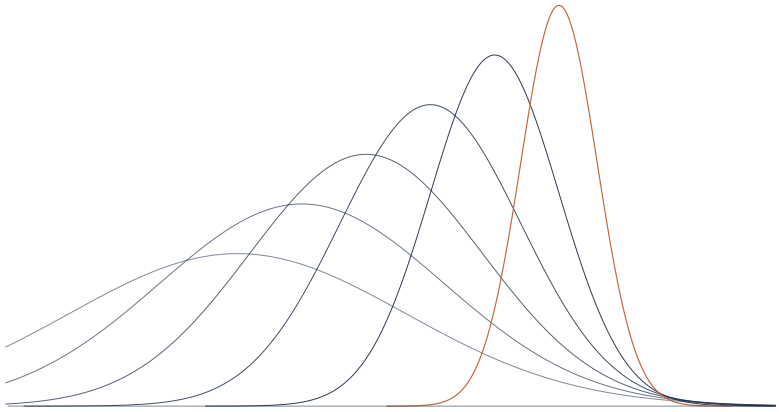
trellis

TRELLIS METHODOLOGY · WHITE PAPER

How Trellis Works



Your ROAS is fine. Your CPA is on target. Your margin tells a different story.

A Catalyst Audit reads what your ad platforms report, reconciles it against your own order record, and applies your gross margin to surface which campaigns are actually profitable after COGS. Every recommendation carries the evidence it rests on and the condition that would reverse the call.

Then the changelog records what was recommended and what was done. The next audit reads its own history, and the recommendations sharpen as your account's record accumulates.

EDITION

v2026-06 · Issued 2026-06-03

trellisads.com/docs/methodology

CONTENTS

Chapter 01

What Trellis is, and what this paper is for

Trellis is a performance marketing audit and trends platform for ecommerce operators running Google Ads, Microsoft Ads, and Meta Ads. It grounds every recommendation in your own first-party order record, your real business economics (gross margin, AOV, COGS, target CPA and ROAS), and a Bayesian statistical core that compounds as your account's history accumulates. The workspace datamart holds your order data. The Changelog holds every Trellis recommendation and every campaign change you record after making it on the ad platforms — Trellis is read-only against those platforms, so the operator is the actor and the recorder. The next audit reads both the datamart and the Changelog before it computes, so the recommendation set gets sharper at your specific account rather than at accounts in general. What the operator gets back is the hours that used to go into dashboards, and a recommendation set defensible enough to act on.

The audit you receive is called a Catalyst Audit. It is built on a small set of named features the rest of this paper explains in turn:

- The datamart is the workspace's first-party order record. It is the source of revenue, margin, attribution validation, and customer-mix signal for every audit.
- Strategic-role classification is how every campaign is sorted (brand, non-brand prospecting, remarketing, shopping, Performance Max, lead-gen, and a small set of others). The classification scopes which benchmarks and recommendations apply.
- Trellis Health Score, the cover-page composite, scores the account against a small set of named pillars (margin, tracking, cadence, context, follow-through). It is profit-weighted, so the audit will not call an account healthy when revenue is up but gross margin is sliding the wrong way.
- The trends view plots the Health Score and its pillars over time, drawing on the datamart's retained history.
- Confidence-gated recommendations carry a tag named in code (Confident, Directional, or Held), so the reader can see, before reading the prose, which findings clear the threshold to act on now and which the audit is sequestering for review.
- Evidence tags mark every number in the report with its provenance: `[FACT]`, `[BAYESIAN]`, `[PROJECTED]`, `[INFERRED]`, `[INSUFFICIENT DATA]`.

- The changelog records every Trellis recommendation and every operator action against the account. The next audit reads it before computing, so recommendations already executed are evaluated against what they actually produced.

A Catalyst Audit gives the reader three concrete things. A ranked, profit-weighted set of findings with named confidence, so the reader can see which findings are ready to act on and which are not. A recommendation set tuned to each campaign's strategic role: a brand campaign is not measured against non-brand expectations. And a recorded trail. The next audit reads it, evaluates what was acted on, and sharpens the next round of recommendations against what the prior round actually produced.

This paper describes the statistical and architectural choices that produce that audit. It is written for the operator who wants to see the mechanism before they trust the output, and for the analyst who wants to verify that the conventions named here are the ones they would have chosen. Worked examples carry an `[ILLUSTRATIVE]` marker on every number.

Chapter 02

What does the work

Every audit starts with the operator's first-party order record, not a generic benchmark. Your datamart holds what your customers actually did: orders, refunds, products, attribution lineage. Your business profile holds the economics that turn revenue into a real decision: gross margin, AOV, target CPA, target ROAS, COGS by SKU. Your Changelog holds every action you took and every recommendation you closed. The audit reads all three before it computes a single number. That is the load-bearing input. The reasoning machinery described below is built around it.

Chapter 03

Four layers, each doing a specific job

Modern recommendation systems run into a familiar problem: a single free-form model can write a confident paragraph about anything, including things the

underlying data does not support. Trellis splits the work across four layers, each with a narrow job.

1. Deterministic spine. Every metric, tier, cohort, and candidate recommendation is computed in Python from your data: KPI grids, profitability tiers, impact dollars, evidence chains, risk scores. Run the same input twice, you get the same numbers twice. Nothing here is generated. It is calculated.

2. Constrained selection. A language model reads the candidate menu produced by Layer 1 and decides which recommendations to surface, in what order, and how to group them by phase. It picks from a fixed list. It cannot author a recommendation that wasn't computed upstream, invent a dollar figure, or alter an evidence tag. Its job is judgment about prioritization, not synthesis of numbers.

3. Templated prose. Each surfaced recommendation renders from a template. Campaign names, dollar figures, evidence tags, and thresholds come from variables computed in Layer 1. The model writes one short paragraph per recommendation — the "why this matters" — bounded by a strict character budget and post-validated.

4. Fail-closed validation. Every dollar figure in the final report traces back to a substrate value or a Python-computed marker. Anything untraceable is rejected and the layer is regenerated. Every recommendation ID traces back to the candidate menu. Every `[FACT]`-tagged claim cross-checks against the data. If validation fails twice, the audit halts rather than ship a number it cannot defend.

The practical effect: two audits run on the same data produce matching campaigns and matching dollar figures. Wording varies. Conclusions do not.

Chapter 04

How we reason

Five principles guide what makes it into an audit and what stays out.

1. **More complexity is not more signal.** A simple rule that holds up under scrutiny beats a sophisticated model that writes a confident paragraph. Trellis uses the smallest method the data needs.
2. **Statistical significance is not practical importance.** A real effect can still be too small to act on. Every recommendation names the effect size alongside the

confidence, so you can see when a real signal is not worth the attention it would cost.

3. **Correlation is not causation.** A pattern in the data is a question, not an answer. The Changelog is where causation is earned: a recommendation closed, a measured outcome in the following window, a delta that holds up under matched comparison.
4. **Data is substrate, not story.** A KPI grid does not tell you what to do; it tells you what is true. The audit's job is to turn what is true into what is worth doing, against your account's economics.
5. **Skepticism is part of the method.** Confidence is earned by data depth, not by tone. When the evidence is thin, the audit names it, and a Held recommendation is a valid result rather than a missed one.

Chapter 05

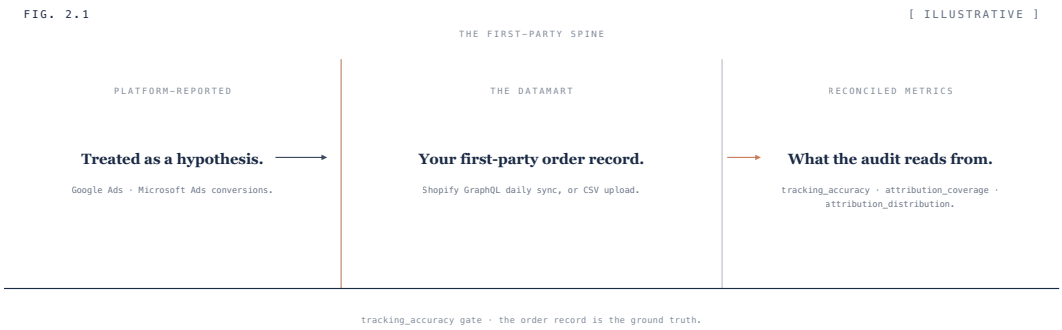
What every recommendation looks like

Every surfaced recommendation pairs an action with a credible alternative the data also supports, and a flip condition that names what would shift the call. The recommendation leads because that is what the data leans toward. The alternative is named because credible-interval reasoning means the parallel hypothesis is real. The flip condition is named so the next audit can hold the recommendation to a falsifiable standard. This is what it means to act on your first-party data rather than on opinion.

Chapter 06

The datamart: your first-party record

Every Catalyst Audit reads from the merchant's own order record, not from platform-reported conversions alone. That order record lives in the workspace datamart, a contextually-aware first-party dataset the audit draws from at every step.



For Shopify merchants, the datamart is populated by a daily GraphQL sync that pulls line-item revenue, unit cost (`productVariant.inventoryItem.unitCost`), customer identity, and UTM-resolved attribution into the workspace. Non-Shopify merchants feed the same schema through a templated CSV upload. Retention is tier-gated: twelve months on Pro, twenty-four months on Agency. The datamart is workspace-scoped; Trellis never blends one customer's order history into another's.

The audit reads from the datamart for revenue, margin (via the COGS join), attribution validation, AOV, repeat-purchase signal, and channel mix. The platform's reported conversion count is treated as a hypothesis the audit reconciles against the order record. The seam between the two is the published `tracking_accuracy` gate. An audit that cannot reconcile platform-reported conversions to the order record inside that gate abstains from margin claims until the gap is named.

First-party data is the merchant's own record of what their customers actually did — collected directly, owned outright, persistent over time. That matters because every other measurement layer in performance marketing has been getting weaker for years.

Browser-side conversion tracking is decaying on multiple fronts: Safari caps first-party cookie lifetimes, Apple's App Tracking Transparency removed device-level tracking access for most iOS users, Chrome is deprecating third-party cookies, and many checkout flows strip URL parameters before the order is recorded. Every click-ID the platform relies on — `gclid`, `msclkid`, `srsltid`, the `gad_*` family — can be missing by the time the order lands; UTM parameters survive when merchants place them in the checkout flow.

The gap between what an ad platform thinks happened and what the merchant's books say happened widens with time. The audit's job is to find that gap, not to inherit it. The customer owns the record. Trellis reads from it.

This paper names the kind of system Trellis uses without exposing the constants that govern it. Tier thresholds, probability constants, prior hyperparameters, control-chart zone boundaries, elasticity constants, and Health Score weights stay in the registry. That calibration is the differentiation.

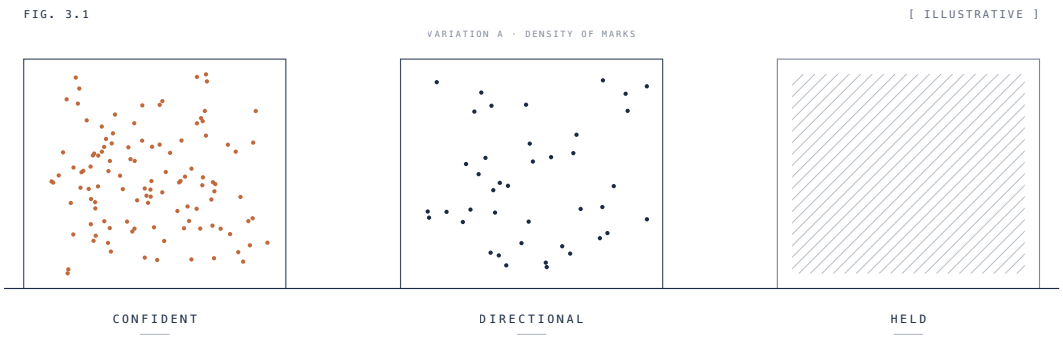
Chapter 07

Confidence: three buckets and the history gate

Some findings are well-evidenced. Others are suggestive but thin. A third group is observational only, held below the evidence threshold. Reporting these three states as a single linear score conflates measurement with abstention. The reader cannot tell whether a low score means *not enough evidence yet* or *evaluated and found wanting*.

Trellis reports confidence in three buckets, following the CONSORT convention, which reports attrition and item-nonresponse separately from any score computed on the evaluable set:

- **Confident** — sufficient evidence to act on now: measured deltas, multiple corroborating signals, a causal story that holds against the alternative-explanation check.
- **Directional** — worth watching: a single signal, a small sample, or a pattern that holds in the audit period but has not yet repeated.
- **Held** — sequestered for review. Below the threshold for impact estimation; the question is recorded, the number is not. Reported separately from the score, not folded into a low-confidence bucket.



Two buckets either flatten the signal or inflate the score. Three preserve the distinction between *evidence below threshold* and *no question worth asking*. A reader scanning the cover sees `14 actionable findings: 14 Confident, 0 Directional` and, separately when applicable, `1 finding held for review: insufficient data to estimate impact`. When every finding is held, the cover reads `Holding all findings for review`, not zero percent.

Bayesian estimation underneath

The per-finding tier is computed using Bayesian estimation. Each metric carries a posterior distribution; the action threshold is a probability (*what is the probability that the true rate exceeds break-even, given the data observed?*) rather than an effect size. The audit uses conjugate prior families for the metrics it estimates: a Beta-Binomial posterior (the standard Bayesian model for yes/no rate questions) for conversion-rate questions, and a Gamma-Poisson posterior (the standard model for count-and-volume questions) for cost-per-acquisition questions. Posteriors begin from uninformative priors, so every account's posterior is built only from that account's data. The posterior tightens as conversion volume grows, which makes the small-account audit honest and the high-volume audit decisive.

The history gate

A campaign with a soft three-day pattern is not a problem. A campaign with a four-week pattern likely is. The audit gates recommendations on the depth of history available, on an ordinal scale of `none` / `thin` / `moderate` / `rich`. First-audit accounts get observational findings, attribution checks, and structural recommendations that hold without longitudinal data. Rich-history accounts get projections, elasticity-bounded budget moves, and pattern-recognition

recommendations that require multiple observation periods. The gate distinguishes *not yet seen enough* from *nothing to see*.

The recommendation carries its own stress test

Every recommendation pairs the recommended action with a credible alternative the same data also supports, and a flip condition that names what would shift the call. The recommendation leads, because that is what the data leans toward. The alternative is named because credible-interval reasoning means the parallel hypothesis is real. The flip condition — a falsifiable threshold — makes the call something the next audit can hold to account. The reader sees the call, the parallel option, and the signal they should watch for to confirm or reverse the decision once they have made it. A recommendation without a stated flip condition is a position the recommender will not be held to; the audit publishes the condition so it can be.

Chapter 08

Strategic role: classifying the campaign before judging it

A brand campaign and a non-brand prospecting campaign are not the same instrument. They serve different intents, draw different traffic, and answer to different benchmarks. Treating them as comparable produces noise; comparing them across that line produces recommendations the operator will recognize as wrong.

Trellis assigns every campaign a `strategic_role` drawn from the canonical PPC naming registry: brand, non-brand prospecting, remarketing, shopping, Performance Max, lead-gen, and a small set of others. The classification is applied at ingest and surfaced in the audit prose. The reader sees the role on every recommendation; the audit's internal scoring keys off it everywhere it matters.

Strategic role changes how the audit reads the data in four places:

- **Section depth.** A non-brand prospecting bucket gets more pages than a brand bucket because there is more room for the operator to move. The audit allocates attention by where decisions live, not by ad-spend share alone.

- **Benchmark choice.** Brand CPA is compared against brand, never against the non-brand mean. Performance Max blended-conversion rate is held to its own standard, not to manual-CPC search.
- **Anomaly thresholds.** The control-chart zones described in the next section are tuned per role. A brand search campaign's normal-zone width is narrower than a Performance Max campaign's, because brand-search volatility is structurally lower.
- **Recommendation phasing.** The audit will not suggest a budget shift from brand to non-brand without naming the role each campaign plays and why the trade-off is being proposed.

FIG. 4.1 [ILLUSTRATIVE]

ROLE-AWARE TUNING

strategic_role ∈ { brand · non-brand prospecting · remarketing · shopping · Performance Max · lead-gen }

The result the reader gets from this classification is recommendations stated in the language they already use. The audit can name "your non-brand prospecting bucket is over-spending relative to brand" instead of "campaign X has a high CPA." The classification is the unit of strategic narrative, not just a code-level tag.

Chapter 09

Anomaly detection: control charts for ad accounts

When the audit looks at a campaign over time, it asks the question Walter Shewhart's control charts were built to answer: *is this campaign in statistical control, or has the underlying process shifted?* Shewhart's separation between common-cause variation (the noise inherent to a stable process) and special-cause variation (a real shift in the process itself) is the canonical way to refuse to act on noise. The audit applies the Western Electric decision rules on top of that framework to classify each metric's state.

How control charts gate the audit [ILLUSTRATIVE]

Each metric is held against three zones tuned to its volatility and to the campaign's strategic role.
The audit leaves the normal zone alone, surfaces the warning zone without acting, and fires only when the metric crosses into action.

A metric in the **normal** zone is left alone, no matter what it did this week. One in the **warning** zone is surfaced but not yet recommended for change. When a metric crosses into the **action** zone, the audit treats the shift as real, and a recommendation fires.

Boundaries are tuned to the volatility of each metric class (CPA, conversion rate, impression share, click-through rate), to the conversion volume of the account, and to the strategic role of the campaign. A high-volume non-brand campaign fires on smaller shifts than a thin-conversion brand campaign where the same shift would be sampling noise. The boundaries live in the statistical registry and are not published.

The point is not to detect every shift. It is to refuse to recommend a change for a shift indistinguishable from noise. Without this layer a recommendation system cannot tell a noisy week from a process shift, and the operator pays for the confusion in churned campaigns and lost spend.

Chapter 10

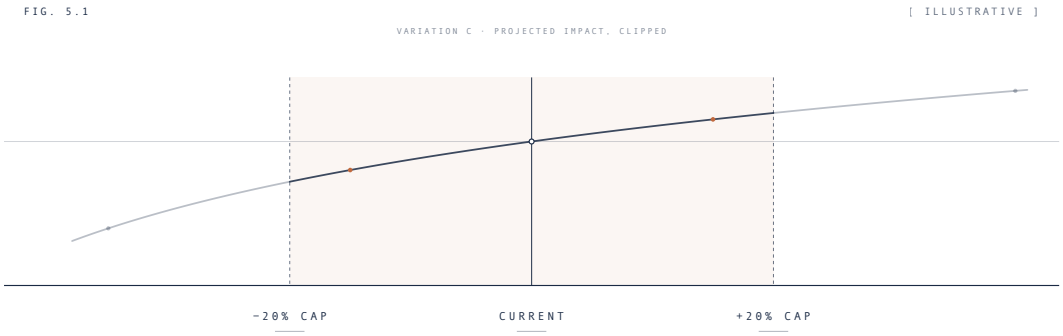
Projecting impact: elasticity within safety caps

When a recommendation includes a dollar impact (*"this change should recover roughly X per month"*), the audit is making a projection. Honest projections require a model that reflects the account's responsiveness and a safety cap that prevents changes the account cannot absorb in a single step.

The model is a constant-elasticity log-log formulation (a percent-to-percent response model: each percent change in spend produces a stable percent change

in conversions) built from the account's own history, drawing on the standard log-log advertising-elasticity convention. An audit-period reset rule bounds the model, so one stale data point cannot warp the projection. Model constants are not published.

The safety cap is non-negotiable. No single recommendation moves a campaign's daily budget by more than ±20 percent. Larger moves split into sequential steps with a seven-day observation gate between them. The cap exists because an uncapped budget-and-bid-strategy change in the audit's pre-history produced a CPC spiral measurable in lost spend before it was caught. The cap is the structural refusal to repeat that mistake.

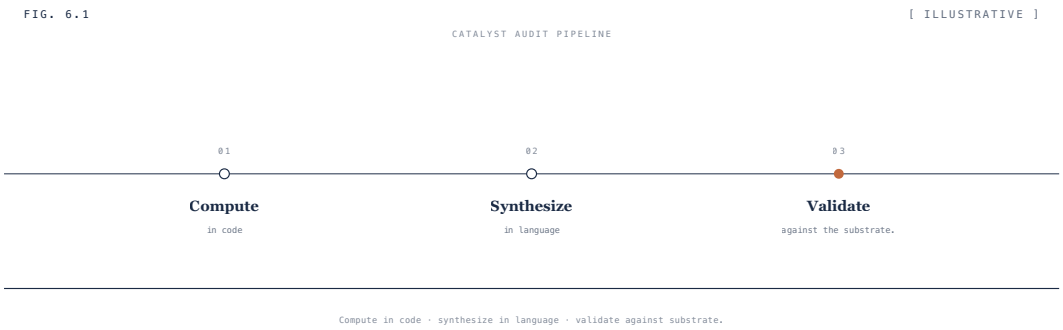


Projections carry an `[INFERRED]` or `[PROJECTED]` tag so the reader can distinguish them from `[FACT]` claims. The audit will not suggest a budget move it cannot bound.

Chapter 11

Compute, synthesize, validate

A language model writes the audit report. It does not compute the audit's findings.



The Catalyst Audit pipeline runs in three stages. **Compute:** structured findings are produced in code (posteriors, control-chart zones, history-gated recommendations, profit-weighted scoring, elasticity-bounded projections). Every numeric claim carries a tagged source. **Synthesize:** those findings are injected into the model's prompt as substrate; the model writes the narrative and organizes the report. **Validate:** post-hoc validators check the returned narrative against the substrate. Every numeric claim is verified; anything that cannot be verified is flagged or held.

Language models write narrative reliably. They are not reliable arithmetic engines. The architecture refuses to ask the model to do the second.

> Compute findings in code. Let the model write the narrative. Validate every numeric claim against the substrate the model was handed.

Evidence tags

Every number carries a provenance tag:

- `[FACT]` — a measured value from the account in the audit window.
- `[BAYESIAN]` — a probability or posterior summary.
- `[PROJECTED]` — a forward estimate with stated assumptions.
- `[INFERRED]` — a pattern conclusion drawn from observable signals.
- `[INSUFFICIENT DATA]` — the audit holds rather than estimates; the reader is told why.

The tags are where post-hoc validators check the model's output, and they tell a marketer whether to scale a finding or hold it. An earlier version of the pipeline asked the model to emit the per-recommendation confidence tier itself. The narrative the model produced could not be anchored back to the underlying

evidence: the tag was inconsistently applied, sometimes contradicted the prose around it, and at times was omitted entirely. The architecture changed. The pipeline derives the tier in code from the evidence tags inside each recommendation's section and appends the stamp after the model returns. Coverage is structural. The code writes the tag, not the model.

Chapter 12

The changelog: bidirectional accountability

The changelog is the feature that turns a single audit into a series. Every recommendation Trellis produces and every action the operator takes against the account is recorded against the workspace, with a date, an author, and a tag from the canonical taxonomy. The next audit reads the changelog before it computes, so a recommendation that was acted on is evaluated against what it actually produced.

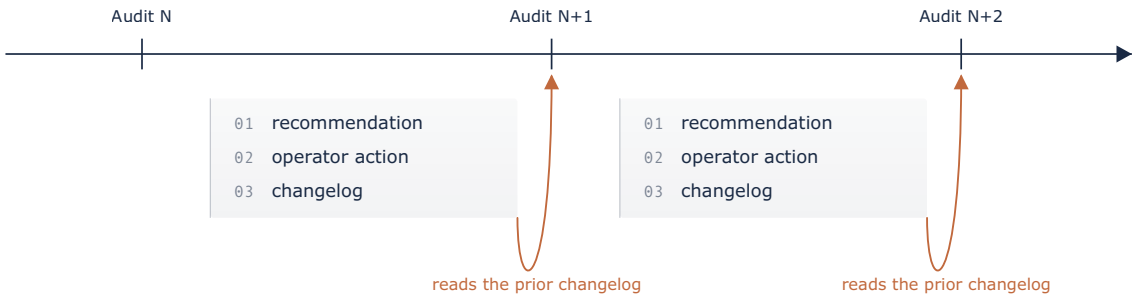
What gets recorded is specific. Trellis writes a changelog entry for every recommendation it has published, with the evidence tag, the projected impact, the strategic role of the affected campaign, and the date of publication.

The operator writes a changelog entry for every account change they make: bid strategy adjustments, budget shifts, structural changes, status flips, new keywords, new negatives, new RSAs. Entries are categorized under a closed-vocabulary taxonomy of twenty-one action verbs and twelve target entities, capped at six tags per entry (`webapp/audits/changelog_tag_taxonomy.yaml`). The vocabulary is closed so the next audit can compare entries across periods without having to interpret free text.

What the operator gets from this is accountability in both directions. Trellis records what it recommended; the operator records what they did; the next audit measures whether the recommendation produced the outcome that was projected for it. Both sides leave a trail. Neither party can quietly walk away from a recommendation that did not work. The audit cites prior changelog entries as evidence when it explains why this audit's recommendation is what it is. The platform audit tabs cannot do this. They show what changed, not what was recommended and whether the change produced what the recommender said it would.

The changelog also tightens the Bayesian posterior described earlier in this paper. A recommendation that has been executed is no longer a hypothesis; it is an observation the next audit reads, and the posterior calibrates to the specific account's responsiveness rather than to a global mean.

How the audit compounds



Posterior tightens with each cycle

wide credible interval

narrow credible interval

Each audit reads the prior audit's recommendations and the operator's actions. The posterior is calibrated by the account's own history.

Three audits in, the model has enough longitudinal history to project budget impact and detect patterns single-period data cannot reveal. Six audits in, it runs full posterior-based inference — calibrated to the account's responsiveness rather than industry averages — and the audit has read what the operator typically acts on and what they hold; the recommendation set sharpens accordingly. The threshold is audit count, not wall-clock time: a weekly cadence reaches six audits in six weeks; a monthly cadence in six months. The cadence sets the speed, the statistical structure is the same. The system gets better at the specific account, not at accounts in the abstract. That is what "Trellis evolves with your account" means in practice.

Every Catalyst Audit ships with a changelog preview, free to every workspace. The full longitudinal changelog is gated to subscribers. The gating is not a paywall on history for its own sake. It is the structural mechanism that makes the compounding loop possible: an account without a retained changelog cannot be re-audited against its own past recommendations, and the posterior cannot tighten beyond what the current audit period sees on its own. Subscription is what turns the audit from a snapshot into a series the reader can hold themselves and Trellis to.

A consequence the reader can verify directly: because every recommendation carries a projected impact and a recorded outcome, the accuracy of Trellis recommendations is itself measurable over time. The audit is auditable on its own work.

Chapter 13

The Trellis Health Score and the trends view

The cover of every Catalyst Audit reports a single Trellis Health composite across a small set of named pillars. The composite is profit-weighted (margin pulls heavier than volume), so the audit will not congratulate an account on growing top-line revenue while gross margin is moving the wrong way.

- **Margin** — profitability against break-even targets.
- **Tracking** — platform-reported conversions reconciled against the first-party order record.
- **Cadence** — whether audits are recent enough that the data behind recommendations is still current.
- **Context** — whether the profile has the data sources (COGS, attribution, business metrics) the audit needs.
- **Follow-through** — whether prior recommendations have been acted on, observed, and recorded.

The composite uses a weakest-pillar surfacing rule. If one pillar is meaningfully below the others, the audit names it and leads recommendation phasing with work that lifts it. The dashboard's next-best-action message comes from this rule. Pillar weights and the composite formula are not published.

A single audit is a snapshot of the account on the day it ran. The Health Score by itself answers *is the account healthy now?* It does not answer the question every operator actually asks, which is *is the account healthier than it was?* That answer needs history.

The trends view is where that question gets answered. It plots the Trellis Health composite and each underlying pillar over time, drawing on the datamart's retained twelve or twenty-four months of order history and the workspace's audit archive. The operator can see whether Margin has been trending up while Tracking quietly deteriorates, whether Follow-through has improved since the workspace

added a second user, whether a strategic-role rebalance from three quarters ago has produced the lift the audit projected for it. None of this is recoverable from a single audit.

The datamart's retention gate is the structural reason the trends view is a paid surface. Without retained first-party history, the trajectory cannot be drawn. A Health snapshot is computed at audit time from the workspace state observed. When a historical snapshot is reconstructed by applying current calibration to an older audit, the trends view marks it backfilled. The reader always knows what was measured at the time and what is being recomputed retrospectively.

Chapter 14

What the audit will not tell you

The audit declines to estimate impact below threshold, project beyond a safety cap, or act on signal indistinguishable from noise. It also declines, on principle, to estimate things outside its read:

- **Creative quality.** Which assets are converting, yes; whether the imagery, copy, or value proposition is working in a deeper sense, no.
- **Audience strategy beyond the data.** Spend-and-conversion patterns, yes; customer-interview or market-sizing research, no.
- **Causal attribution beyond connected channels.** Platform conversions reconciled to first-party orders, yes; view-through, dark social, or unconnected channels, no.
- **Decisions the operator is best placed to make.** Product launches, pricing, rebrands: the audit reports the data; the operator makes the call.
- **Patterns a single audit period cannot evaluate.** Some signals require multiple audits to surface. The history gate makes this trade-off explicit.

The audit reports what it can defend and stays quiet about the rest. What it does cover is published: the ±20 percent budget cap, the `tracking_accuracy` gate, the evidence-tag vocabulary, the strategic-role taxonomy, the three-bucket confidence reporting, and the Health Score pillars. The constants that calibrate the audit (tier thresholds, probability constants, prior hyperparameters, control-chart zone boundaries, elasticity constants, Health Score pillar weights) stay in the registry. The conventions the methodology draws on are named at point of claim throughout this paper: CONSORT reporting for confidence, Shewhart control charts with Western Electric rules for anomaly detection, conjugate Beta-Binomial

and Gamma-Poisson Bayesian estimation with uninformative priors, and the constant-elasticity log-log family for advertising-response projection.

The methodology is named. The recommendations that follow are the operator's to act on.