

Telescoping Decision-PGA: Seeing The Smaller Shape Inside The Larger One

Zachary D. Michels, PhD

May 22, 2026

This companion perspective extends [Decision-PGA](#) and the [Need for Decision-State Diagnostics](#) by focusing on how uncertainty clouds can contain smaller local structures and cross-document bridges. A PDF copy is available at [assets/telescoping-decision-pga.pdf](#).

A Diagnostic Should Be Able To Zoom

Workflows rarely fail in one clean motion. More often, they hesitate.

A document extractor may recognize that a table row matters, but not know whether the row is a parent specimen, a child aliquot, a timing note, or a formatting fragment. A routing agent may know that it needs more evidence, but not whether the missing evidence is in a source document, a standards table, a user preference, or a downstream system constraint. A reviewer may see "low confidence," but low confidence is not yet a diagnosis. It is closer to a warning light.

Decision-PGA starts from a modest idea: when a system repeatedly estimates support across a set of possible decisions, those estimates form a shape. The shape may tell us something about the kind of uncertainty the system is facing.

The useful next step is to let that shape zoom.

A broad uncertainty cloud may first suggest, "this is a layout problem." That is better than saying "low confidence," but it may still be too broad to act on. Inside that layout cloud there may be smaller patterns: row-boundary confusion, visual-marker binding, child-output grouping, page-continuation ambiguity, or false row fragments. Each smaller pattern suggests a different next move.

The point is not only to pick the final answer. The point is to help the workflow choose its next small, useful action.

What Decision-PGA Borrows From PGA

Principal Geodesic Analysis generalizes the intuition of PCA to nonlinear spaces. In shape analysis, that matters because some objects cannot be averaged or decomposed cleanly with ordinary straight-line geometry. Decision-PGA borrows the same caution: do not flatten a curved object too early if its shape contains useful information.

In Decision-PGA, the object is not an anatomical shape. It is a probability-like observation over a candidate decision set. The candidate set might contain possible actions, routes, evidence interpretations, or values. Repeated observations over those options become points on a probability simplex. With a square-root transform, those points can be viewed on a positive unit sphere under Fisher-Rao geometry. The system can then estimate a mean, map observations into a local tangent space, and inspect the main directions of dispersion.

That is the mathematical frame. The practical question is simpler:

What shape does the uncertainty have?

A tight cloud near one action suggests stability. A stretched cloud between two actions suggests binary ambiguity. A broad cloud suggests diffuse uncertainty. A cloud sitting near a threshold suggests boundary sensitivity. A cloud that moves over time suggests a regime shift.

Those first-order states are useful, but they are not the whole story.

The Cloud May Contain Smaller Clouds

A good label can still be too large.

Suppose a document extraction system repeatedly routes a problem to "row-band geometry." That is already helpful. It tells the developer to look at spatial layout rather than synonyms, standards mapping, or human review. But row-band geometry is not one problem. It is a family of smaller problems.

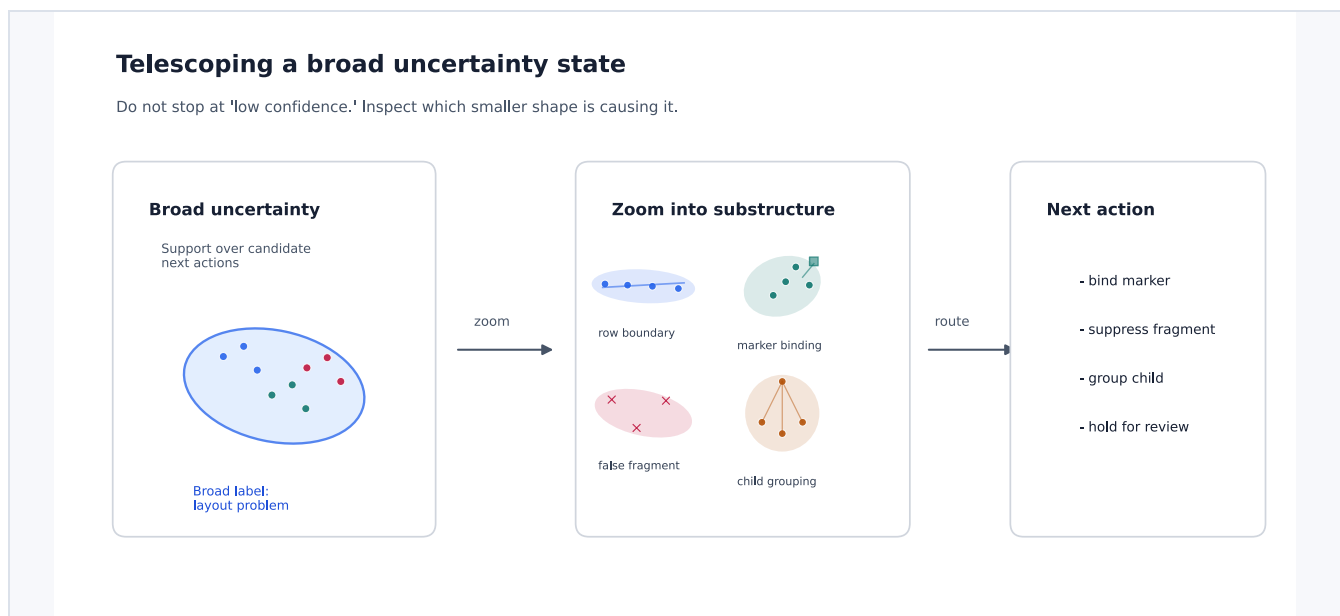
One geometry cloud may be driven by a visual quantity marker in a "What to Ship" region. Another may be driven by OCR fragments that look like specimen rows but are only table debris. Another may involve a real specimen label whose operational details are spread horizontally across neighboring cells. Another may involve a child row that inherits a parent label vertically.

All of those cases can live inside the larger geometry cloud. If the workflow stops at the broad label, it may keep applying a plausible but blunt strategy. If it telescopes into the cloud, it can ask the more useful question:

What kind of geometry problem is this?

That second question changes the action. A quantity-marker problem suggests binding a marker to a row band. A false-fragment problem suggests suppression. A parent-child inheritance problem suggests grouping. A partial table-authority problem suggests preserving review status while collecting more evidence.

The diagnostic did not merely select an action. It helped define a better action vocabulary.



Telescoping a broad uncertainty state. Telescoping Decision-PGA treats a broad uncertainty cloud as a place to inspect, not just a label to assign. A broad layout cloud can be opened into smaller local substructures, and each smaller substructure suggests a more specific next action.

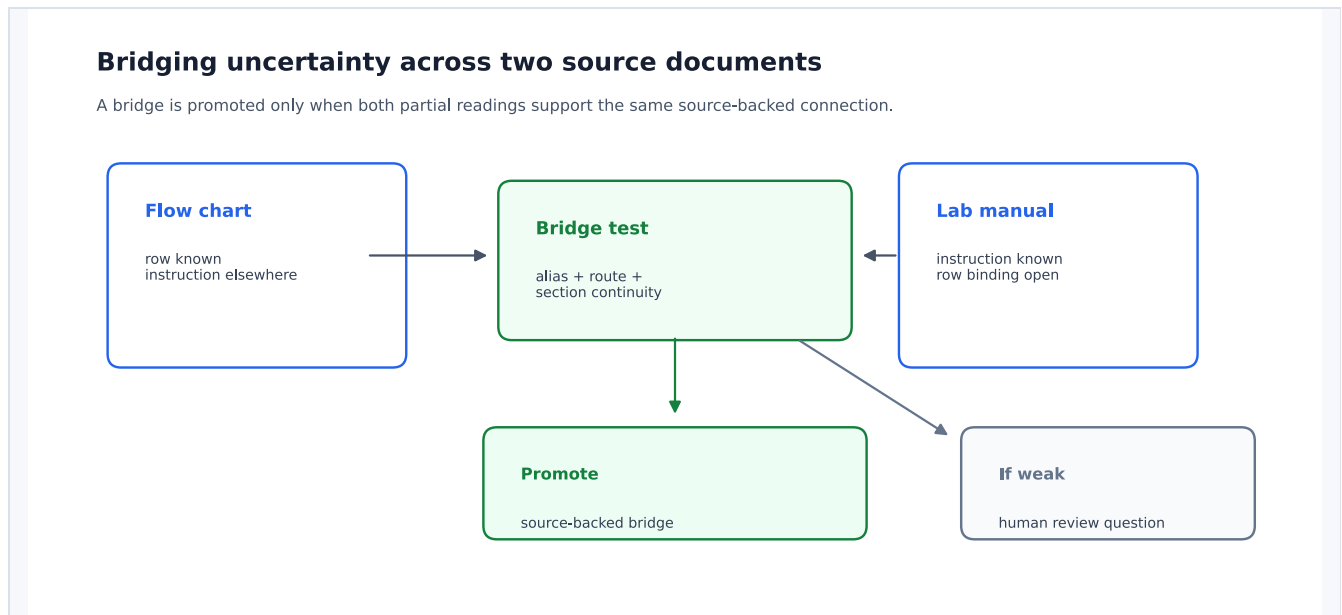
The Bridge Case: When Two Uncertainties Belong Together

Some problems cannot be resolved by looking at one document in isolation.

Imagine a flow chart cell that says, in effect, "refer to the lab manual for processing instructions." The flow chart creates one uncertainty: it identifies the specimen activity, but the instruction itself is missing from that page. The lab manual creates a different uncertainty: it contains processing instructions, but the extractor must decide which activity those instructions belong to.

Handled separately, both uncertainties may look incomplete. The flow-chart side looks like a missing fact. The lab-manual side looks like an alias or row-binding problem.

The bridge appears when both uncertainties are projected into a shared diagnostic space. The flow chart has a specimen label, routing context, and a pointer to another document. The lab manual has an instruction block, an activity name or alias, and nearby processing context. If those pieces align strongly enough, the workflow can join them as source-backed evidence. If they align weakly, it should preserve the bridge as a review question rather than guessing.



Bridging uncertainty across two source documents. A bridge is not a guess. It is a testable connection between partial readings. The flow chart may know the row but not the instruction; the lab manual may know the instruction but not the row binding. Only when the shared context is strong enough should the workflow promote the source-backed bridge; weaker links stay a human review question.

This is one of the places where telescoping is especially useful. The broad label might be "source gap." The smaller local structure might be "cross-document bridge needed." Inside that bridge problem, the next candidate actions might be: normalize aliases, check section continuity, compare processing context, bind to a specimen row, or ask for expert confirmation.

The bridge does not make uncertainty disappear. It makes the uncertainty inspectable and actionable.

Telescoping As A Workflow Pattern

Telescoping Decision-PGA can be treated as a repeated diagnostic loop:

1. Define the candidate decision set at the current scale.
2. Collect repeated probability-like observations.
3. Characterize the uncertainty shape.
4. Route to an action or to a smaller diagnostic space.
5. Re-observe after the action.
6. Preserve the lesson as a state, feature, test, rule, or review question.

At the macro scale, the candidate decisions might be extract, retrieve, clarify, map, review, or defer. Inside "extract," the candidate decisions might become bind visual marker, reconstruct split row, suppress table fragment, join lab-manual alias, or flag missing source fact. Inside "bind visual marker," the candidate set may narrow again to same-line marker, nearby visual band, page-continuation marker, inherited child output, or ambiguous icon.

This is not recursion for decoration. It is a way to avoid forcing different failure modes into the same broad label.

Many expensive agentic failures come from repeated use of a strategy that was locally plausible but not specific enough. A system keeps searching when it should ask. It keeps mapping when it should retrieve. It keeps applying synonyms when the problem is spatial. Telescoping diagnostics are meant to interrupt that pattern.

Learning By Filling In The Shape

The value appears over iterations.

At first, a workflow may only know a few broad uncertainty states: stable, ambiguous, diffuse, boundary-sensitive, drifting. After several development cycles, it may learn that "boundary-sensitive" splits into smaller substructures. Some boundary cases are threshold artifacts. Some are missing source facts. Some are layout-binding problems. Some are standards translation problems. Some are cross-document bridge problems.

Each resolved case adds structure to the diagnostic map.

That learning does not have to mean silent retraining. It can be auditable. A failed extraction becomes a labeled fixture. A labeled fixture becomes a test. A test-backed fix becomes a decision-state transition. A decision-state transition becomes part of the next report. Over time, the system accumulates a map of where its uncertainty tends to live and which actions have historically reduced it.

This matters because agentic efficiency is not only about choosing the final answer faster. Much of the gain comes from making micro-actions less wasteful.

A diagnostic that prevents three irrelevant searches is useful. A state label that tells the extractor to inspect row geometry instead of tuning synonyms is useful. A substructure label that distinguishes visual child-output binding from false row fragments is useful. These are not dramatic decisions, but they compound.

Why This Is More Than Action Selection

It is tempting to describe Decision-PGA as a next-action selector. That is partly true, but too narrow.

The more interesting role is observability. Decision-PGA gives a workflow a way to inspect its own task-facing uncertainty. It does not require access to hidden model cognition. It does not need to claim that all reasoning lives on a smooth manifold. It only needs repeated probability-like observations over a defined candidate set.

That modest boundary is a strength. The method can be attached to document extraction, tool routing, retrieval conflict, standards mapping, feedback triage, review queue prioritization, or multi-step workflow monitoring. The labels change, but the central question remains:

What does the support cloud look like, and what kind of action does that shape suggest?

At one scale, the cloud helps select a route. At another scale, it helps define a better route vocabulary. At a still smaller scale, it helps identify which local uncertainty patterns deserve tests, metrics, or human review.

Decision-PGA is not pretending to solve the domain. It is helping the system see the shape of its own work.

Cautions

The geometric framing should stay disciplined.

PGA-style diagnostics should be compared against simpler signals: entropy, margin, agreement, calibration, switch rate, clustering, and ordinary trajectory metrics. If those simpler tools explain the useful distinction, that should be reported. The point is not to decorate workflow telemetry with geometry. The point is to find cases where dispersion shape carries operational information that simpler summaries flatten away.

The candidate set also matters. A poor candidate vocabulary produces poor geometry. If the system only offers "accept" and "review," the cloud cannot reveal whether review is needed because of missing source context, ambiguous layout, conflicting standards, or unsafe dropdown mapping. Telescoping helps because broad labels can become smaller local vocabularies once the workflow knows where to look.

Finally, a diagnostic state is not truth. It is a routing signal. It can make review more focused, testing more systematic, and automation more cautious. It does not remove the need for source authority, validation, domain standards, or human accountability.

What To Build Next

A useful next step is a lesson ledger for agentic workflows. Each entry would record:

- the observed cloud shape;
- the local substructure, if any;
- the action attempted;
- the evidence used;
- the metric movement after the action;
- whether the lesson was promoted, held for review, or rejected.

That ledger would make learning visible. It would show which uncertainty shapes repeatedly respond to which actions. It would also show where the method does not help.

For document extraction, such a ledger might reveal that some "layout" clouds usually become visual-marker binding problems, while others become row-fragment suppression problems. For retrieval, it might show that some diffuse clouds resolve after one missing source is added, while others remain diffuse because the candidate action set is too coarse. For mapping, it might separate true standards gaps from cases where the source fact was already present but expressed in a local vocabulary.

The ledger is how the framework can become more than a dashboard. It becomes a way for the system to refine the task manifold it is operating on.

Conclusion

Decision-PGA is not only a tool for choosing between actions. It is a way to describe uncertainty shape in a form that a workflow can use.

The telescoping version extends that idea. A broad decision cloud can be examined for substructure. That substructure can define smaller candidate spaces. Those smaller spaces can suggest more precise actions. The results of those actions can feed back into the next diagnostic pass.

In that loop, learning is not mysterious. It is the gradual filling-in of shape: first the large domains, then the local ridges, boundaries, bridges, and wells that explain why one small action helps and another does not.

That may be one of the important requirements for practical agentic systems. They need to answer questions, but they also need to learn how to move through work. Decision-PGA offers one modest, inspectable way to help them do that.