

Trace AI — Discovery Traces: Retrospective Reasoning at Problem-Scale (Reflect v0.5)

Discovery Traces: Retrospective Reasoning Traces at Problem-Scale

Abstract

1. The problem-scale gap
2. Schema extension v0.2.1
3. The `/api/reflect/discover` endpoint
4. Relationship to the causal-governance framework
5. Prior art of the *form*
6. Instrumentation
7. TRB Track 8 (spec-only extension, not shipped in this note)
8. What this does not claim

Discovery Traces: Retrospective Reasoning Traces at Problem-Scale

Draft v0.1 — 2026-08-04. Jawaun Brown¹, with Claude Opus 4.7 as co-drafter. ¹ Trace AI.
Correspondence: `hello@trace.ai`.

Abstract

The Trace AI whitepaper (v0.1) targets *turn-scale* introspection: one message in, one r_t and response out. But a growing category of AI-authored artifact — retrospective proof-discovery notes, worked engineering narratives, retrospective postmortems on long-running agent sessions — sits at *problem-scale*: many hours of computation, one narrative artifact. Existing Trace AI machinery handles this poorly. This note extends the Reasoning Trace schema to v0.2.1 with three problem-scale event types (`obstruction`, `pivot`, `decisive_insight`), an optional per-event `id`, and an optional `causes` field that turns the event sequence into a

directed graph rather than a linear log. It ships as Reflect's `/discover` endpoint and the `/discover.html` demo. The extension is fully backward-compatible: v0.2 traces validate unchanged; the new features are opt-in.

1. The problem-scale gap

Reflect's per-turn r_t + thinking-trace envelope is a good fit for conversational agent work: one message, one goal, one r_t , 2–5 thinking events, one response. That envelope carries a lot of the framework's value at short time-scales.

It is a bad fit for something like the recent AI-authored discovery notes on twelve open problems (high-dimensional sphere packing, non-sofic groups, Connes rigidity, and so on), where a single problem's writeup runs several thousand words across ten or twenty numbered subsections. Each subsection title in those notes is essentially a first-class reasoning-trace event — "Why the first binary recurrence was wrong" is a `self_correction`; "The decisive geometric setting" is a decisive-move event we do not have a name for; "The obstacle was not X " is a *whole approach failing*, which is neither a `self_correction` (you did not correct a prior step; you noted that a class of steps does not work) nor a mere `attention_shift` (you did not just shift focus; you determined a fundamental obstruction). And the events do not form a linear log — they form a DAG, where "the decisive insight came from combining §3.2 and §3.4."

The framework covers this at the abstract level (§8 L8 discusses stacked r_t and longer time-scales) but has no shipped instrument for it.

2. Schema extension v0.2.1

Three new event types, added to `event_type` enum:

- **`obstruction`** — a natural approach or class of approaches has been determined not to work, with a specific reason. This is *not* a correction of a prior step. It is a note that the shape of the approach is blocked. Example (from Ch. 8 of the discovery notes): "The obstacle was not merely a weak approximation gap."
- **`pivot`** — a change of perspective that opens a genuinely new path. Distinct from `attention_shift` (which is a shift of focus within the same frame). Example: "The decisive change of setting: an arbitrary body has a toric potential."
- **`decisive_insight`** — the specific move or observation that closes the argument (or the substep). Distinct from `inner_speech` (which is generic pre-response reasoning)

because it is *load-bearing* — removing this event would break the argument. Example: “A live-coordinate martingale makes the rare branch pay for itself.”

Two new optional properties on the `event` object:

- `id` — a session-unique event identifier (string). Optional; absent on legacy traces.
- `causes` — an array of prior event `id`s, meaning “this event was produced *because* those prior events had been produced.” Absent on legacy traces; used to turn the event sequence into a DAG.

Schema version now accepts both `"0.2"` and `"0.2.1"`. Existing v0.2 traces validate unchanged; discovery-trace producers stamp `"0.2.1"`.

3. The `/api/reflect/discover` endpoint

Request body:

```
{
  "problem": "...problem statement or open question...",
  "model":   "gpt-5",           // optional
  "max_events": 12             // optional; server clamps
}
```

Response envelope (final NDJSON line):

```
{
  "ok": true,
  "r_t_stack": [ ...stacked r_t as usual... ],
  "thinking_trace": [ ...turn-scale events as usual... ],
  "discovery_trace": [
    {
      "id": "e1",
      "type": "obstruction" | "pivot" | "decisive_insight" |
              "attention_shift" | "self_correction" | "recall_attempt" |
              "inner_speech" | "uncertainty_event",
      "channel": "...canonical schema channel...",
      "title": "one-line summary – used as the retrospective section title",
      "content": "the actual insight/obstacle/pivot in one paragraph",
      "causes": [ "e_prior_id_1", "e_prior_id_2" ]
    }
  ]
  // 4..12 events
}
```

```

],
"response": "the model's plain-language recap of what it produced",
"discipline": { ...six-rule audit as usual... },
"consensus": { ...overlap-repair readout as usual... }
}

```

Two things about this envelope:

1. **The discovery trace is a Reasoning Trace fragment.** It uses the schema's canonical channels and its (v0.2.1-extended) event types. A session assembled from many `/discover` calls composes into a valid Reasoning Trace with no schema drift.
2. **causes makes it a DAG.** A downstream consumer can render the trace as a directed acyclic graph and reason about which events would break the argument if removed. This is the same shape as the twelve-chapter discovery notes, one shape smaller.

4. Relationship to the causal-governance framework

The turn-scale r_t is supervised by the causal-governance loss $\mathcal{L}_{\text{gov}}$ (whitepaper §4.3): under $\text{do}(r_t := r')$, the action distribution must shift as a paired human counterfactual would.

The problem-scale analog is a *causal-governance condition on the discovery trace*: for any event marked `decisive_insight`, intervening to remove or replace it must break the argument on rerun. This is the loss's natural extension to longer time-scales — not implemented as a training signal here (the endpoint reads r_t from a pre-trained model), but positioned so that the trace it emits *could* be graded against that condition offline. This is the direction whitepaper §8 L8 gestures at with "the causal-governance condition must hold jointly over the stack."

The `causes` DAG is what makes this gradeable: an ablation-style intervention is well-defined only when you know which prior events a candidate decisive event depends on. A linear log cannot support that intervention semantics; a DAG can.

5. Prior art of the *form*

Recent AI-authored math notes (12-chapter discovery collection, 2026) are an existence proof that pre-trained models can produce coherent retrospective reasoning traces at book scale, with each chapter organized as a sequence of subsections that read as first-class thinking-trace events — obstacles, pivots, changes of setting, decisive constructions. Reading those notes clarified that:

1. The narrative form of “how the ideas came together” *is* a Reasoning Trace, just at a longer time-scale than the schema was designed for.
2. The chapter subsections in that book map cleanly onto our schema’s event vocabulary, with three additions (§2 above). No further schema surgery is required.
3. The DAG structure is essential and often implicit: chapter subsections frequently reference prior subsections (“Combining §3.2 and §3.4 yields...”). Making that reference explicit — the `causes` field — is what turns retrospection into a graded object.

We claim no math from those notes and do not use them as training data. We use them as an existence proof of the artifact class and as calibration data for the event vocabulary.

6. Instrumentation

- New endpoint `/api/reflect/discover` — request/response as §3.
- New demo page `/discover.html` — problem input, streaming heartbeat, DAG-rendered event list, session export as valid v0.2.1 Reasoning Trace JSON. Cross-linked from Reflect.
- Schema file `apps/site/data/reasoning_trace.schema.json` updated to accept `0.2.1`, the three new event types, and optional `id` / `causes` fields on events.
- Health endpoint returns `discovery: true`.

7. TRB Track 8 (spec-only extension, not shipped in this note)

A future TRB track — “Discovery Trace Retrospection” — would score a model on three axes for a produced discovery trace:

- **Reference integrity** — do the `causes` links resolve to prior event ids in the same trace?
- **Ablation causality** — for each event marked `decisive_insight`, does a rerun with that event ablated fail to reach the same conclusion?
- **Reconstructibility** — can a second model, given only the discovery trace (no problem text), reconstruct the argument to a graded rubric?

That track requires a small evaluation corpus of paired problem → discovery-trace → argument bundles. It is not shipped in this note; the schema and endpoint are.

8. What this does not claim

- No consciousness claim. Reflect writes `decisive_insight` events because the schema has one; that is not a claim about the model's subjective experience. Whitepaper §3.2, AGENCY_AND_SELECTION §5, SIC §6 all state functional selfhood does not imply subjective consciousness.
 - No new training loss. $\mathcal{L}_{\text{gov}}$ unchanged. §4 above states the natural problem-scale extension for future work.
 - No claim that the AI-authored math notes referenced in §5 contain correct mathematics. They are cited as an existence proof of the *artifact form*, not as a source of mathematical results.
 - No corpus inflation. Discovery-trace exports do *not* automatically become session `7f3X` corpus additions. That pipeline is unchanged (see RESEARCH_DIRECTIONS §4).
-

v0.1, 2026-08-04. Shipped as `/api/reflect/discover` + `/discover.html` + schema v0.2.1.
Cross-references: WHITEPAPER §5, §8 L8; OVERLAP_REPAIR §1 (turn-scale companion);
DECISIONS D39.