

Optional Oracles in Excadrill

With a focus on the deriver and dreamer

Eugene Eisenstein

Plastic Labs

July 22, 2026

What?

Oracles for performance attribution

- We need a way to attribute benchmark performance to individual Honcho system components.
- Today we can ablate the individual components.
- Here we enable also replacing them with an oracle.
- Comparing oracle-substituted runs isolates each component's headroom, which is arguably more interesting than ablation.
- In this document we focus on the deriver and dreamer components. The dialectic can follow a similar pattern but it is simpler - it is briefly mentioned.
- We treat all these as black-box components and don't consider the possibility of mocking their tool calls.

How?

High-level implementation

- **Deriver & dreamer oracles:** inject documents straight into the DB that mimic what these stages would produce, but sourced from the oracle.
- **Dialectic oracle:** several meanings here:
 - Inject `final_evidence` inside Excadrill itself from the oracle - this is essentially a benchmark ceiling sanity check.
 - Inject `golden_conclusions` inside Honcho into the dialectic - this is essentially the dreamer oracle, up to retrieval, which is a tool with a query.
- **Summarization is untouched:** it always runs as normal.
- The oracle data can be provided by humans but it can also be generated by a model, which scales much better.

Oracle data generation strategy

- Default oracle model: **Opus 4.8**, given the problem text *and the solution*, producing structured output:

```
{  
  "extracted_facts": list[str],  
  "derived_facts": {  
    "deductive": list[str],  
    "inductive": list[str]  
  }  
}
```

- -extraction-only — restrict the dialectic oracle to derive-like extracted facts only.
- Generators are instrumented with LangFuse and track token usage as they run.
- Every example yields golds (the facts needed for that example) plus a large set of *distractors* — valid memories that are irrelevant and generated from other parts of the benchmark — so the dialectic is still genuinely exercised.
- Each oracular conclusion is tagged **explicit**, **deductive**, or **inductive**.

Oracle data generation, per benchmark (1/2)

- Always instructed to produce decontextualized and minimal extracted facts with short prompts explaining this, pulled from Honcho prompts.

BEAM

- **Gold:** the LLM-as-judge grading rubric grounds which facts are needed, also conversation references exist for everything except abstention.
- **Distractors:** all oracular conclusions for the entire conversation.

LoCoMo

- **Gold:** the `evidence` key gives direct conversation sourcing.
- **Distractors:** all oracular conclusions for the entire conversation.

Oracle data generation, per benchmark (2/2)

LongMemEval

- **Gold non-abstention:** haystack_session_ids identified as answer_session_ids with turn-level has_answer marked when rendered for the oracle LLM;
- **Gold abstention:** benchmark-provided “near-miss” answer_session_ids and a separate prompt to the oracle that these are abstention near-misses. Deductive and inductive conclusions are always excluded.
- **Distractors:** randomly sampled turns from session_ids outside the answer_session_ids golds.

Sample oracle data - BEAM

BEAM

```
{
  "content": "The user is developing a personal budget tracker application using Flask.",
  "level": "explicit"
},
{
  "content": "The user is implementing three security features across sessions: password hashing, role-based access control, and account lockout after failed login attempts.",
  "level": "deductive",
  "source_ids": [
    "1-e0",
    "1-e1", ...
  ]
}
```

Sample oracle data - LoCoMo

LoCoMo

```
{
  "content": "Caroline went to an LGBTQ support group.",
  "level": "explicit",
  "observer": "Caroline",
  "observed": "Caroline"
},
{
  "content": "Caroline's educational path is aimed at qualifying her for mental health
            support work.",
  "level": "deductive",
  "source_ids": [
    "conv-26-2-e0",
    "conv-26-2-e1", ...
  ],
  "observer": "Caroline",
  "observed": "Caroline"
},
```

Sample oracle data - LongMem

LongMem

```
{
  "content": "Considering taking acting classes focusing on scene study and character
              development.",
  "level": "explicit",
  "origin": "gold"
},
{
  "content": "Has a personal interest in theater and acting, both as an audience member
              and aspiring performer.",
  "level": "inductive",
  "source_ids": [
    "58bf7951-e0",
    "58bf7951-e1", ...
  ],
  "pattern_type": "tendency",
  "confidence": "high",
  "origin": "gold"
}
```