@demotoprod · EVALS · also LEVERS

Determinism checklist

In a September 2025 lab test, one prompt sent 1,000 times at temperature zero to an open 235B-parameter model returned 80 unique completions. Batch size changes the order of the arithmetic, and server load sets the batch size. Five checks, about an afternoon each, make your tests, logs and answers repeatable.

1 Score fields, never the exact string

Of the lab test's 80 unique completions, the most common appeared 78 times in 1,000. In a study of five models set to be deterministic, accuracy varied by up to 15% across runs, and parsed answers were often more stable than raw text.

- Replace exact-string asserts with checks on parsed fields: labels, numbers, IDs. Ignore wording.
- Run each case 10 times; report the lowest and highest score; fail the build on the lowest.

2 Pin the model ID and log it with every run

One provider documents that a model ID is a pinned version and an alias resolves to the most recent dated snapshot, and that infrastructure updates can still change behaviour slightly. Another returns a fingerprint to flag backend changes, yet warns of variation when seed and fingerprint match.

- Replace every alias in config with a pinned model ID; add a test that fails if an alias appears.
- Log model ID, seed, parameters and fingerprint (if returned) on every call; alert when it changes.

3 Cache by input hash where answers must not move

Provider prompt caching saves key-value tensors for a repeated input prefix, not finished answers; new input is still processed to generate a new response. Hashing needs an invariant input format to be repeatable, which a canonical JSON form gives.

- Hash the canonical request (model ID, parameters, full prompt); on a hit, return the stored answer.
- A new model ID makes a new key; keep old entries as the record of what was answered.

```
import hashlib, json
def cache_key(model_id, params, prompt):
    body = json.dumps([model_id, params, prompt], sort_keys=True,
                      separators=(",", ":"), ensure_ascii=False)  # close to RFC 8785
    return hashlib.sha256(body.encode("utf-8")).hexdigest()
```

4 Replay one prompt 100 times, count the answers

In the lab test, 1,000 completions of 1,000 tokens at temperature 0 gave 80 unique answers, all identical for 102 tokens and split at token 103. One provider's guide says a larger max_tokens generally makes responses less deterministic, even with a seed set.

- Replay your 5 longest production prompts 100 times at temperature 0; count the unique outputs.
- Note where outputs first differ; repeat after any change of model ID or provider.

```
outs = [ask(prompt, temperature=0) for _ in range(100)]   # ask: your own client call
print(len(set(outs)), "unique answers in", len(outs), "runs")
```

5 Switch on batch-invariant kernels if you host the model

With batch-invariant normalisation, matrix-multiply and attention kernels, all 1,000 completions were identical, and a 1,000-sequence job on an 8B model took 42 s, up from 26 s. One serving engine documents such a mode as beta, needing GPUs of compute capability 8.0 or higher.

- Enable the mode on staging; replay the check 4 prompts under load and expect 1 unique output.
- Time a representative job before and after; keep it on only where an answer must not move.

Nitish Gautam · indica-tech.com · hello@indica-tech.com

INDICA TECH · @DEMOTOPROD

Nitish Gautam

Founder-led AI engineering. Demo to production, under NDA, in regulated industries, on fixed scope.

Paste into your project

A DETERMINISM POLICY TO KEEP BESIDE YOUR EVALS

```
# counts are planning estimates; set your own
model_id: <pinned model ID, never an alias>
temperature: 0
log_every_call: [model_id, seed, params, fingerprint]
eval:
  runs_per_case: 10          # fail the build on the lowest score
  assert_on: parsed_fields   # labels, numbers, IDs; never exact text
replay:
  prompts: 5                 # your longest production prompts
  runs: 100
  rerun_on: [model_id_change, provider_change]
cache_key: sha256 of canonical JSON [model_id, params, prompt]
```

Where this list comes from

Ten-plus years taking products from zero to one across seven sectors, including defence, fintech, medical and telecom, with AI shipped into regulated and air-gapped environments. The @demotoprod channel takes one production AI failure apart each day. These checklists are the same work written down: every item is a dated failure or a documented default, with the control that stops it and the primary source.

If this list found a gap

1. ### Your demo works and production is stuck.

   Production Readiness Audit. Fixed price, two weeks. A five-dimension scorecard, cost and failure analysis, and a 90-day execution roadmap.

2. ### You need senior AI judgement without a full-time hire.

   Fractional AI Lead. Monthly: 90-day minimum, then 30 days' notice either side. Weekly 1:1 with the founder or CTO, architecture and code review, technical hiring support, investor-ready due diligence.

3. ### An AI feature has to work, by a date.

   AI Build Sprint. Fixed price, four to eight weeks. Scope signed off up front, weekly demos, CI/CD and observability included, 30-day production warranty.

4. ### A regulator, an auditor or a customer's security team is asking.

   AI Governance & Compliance. Fixed price, four weeks. ISO 27001 / SOC 2-aligned controls, a model risk framework, human-in-the-loop and audit logging, regulator-ready documentation.

BOOK A CALL
indica-tech.com/#contact

EMAIL
hello@indica-tech.com

WEBSITE
indica-tech.com

ALL CHECKLISTS
indica-tech.com/checklists/

YOUTUBE
youtube.com/@demotoprod

LINKEDIN
linkedin.com/in/startup-founderx

INSTAGRAM
instagram.com/demotoprod

TIKTOK
tiktok.com/@demotoprod

Free to share. The live list, with every source link: indica-tech.com/checklists/#llm-determinism