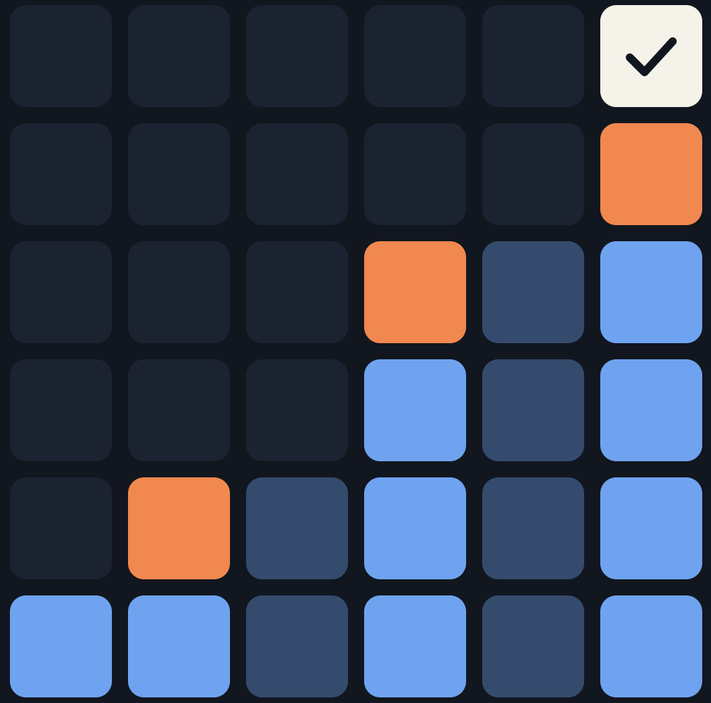
POST-TRAINING TUNING AND CONTINUOUS LEARNING

The system is built to keep learning after it ships.

Small open models are tuned on checked examples, and a rule fixed before scoring decides whether one replaces its base.



Why training alone doesn’t fix it

01

A model freezes when it ships

It never sees the tools, schemas and failures that appear after its release, so the same mistakes repeat.

02

Bigger isn’t the fix

On the same frozen test, Llama 3.2 3B beat the 1B, 52.0% to 12.0%, yet wrote typed values as strings in 17.5% of its calls, against 8.3%.

03

Even a 30B model misses

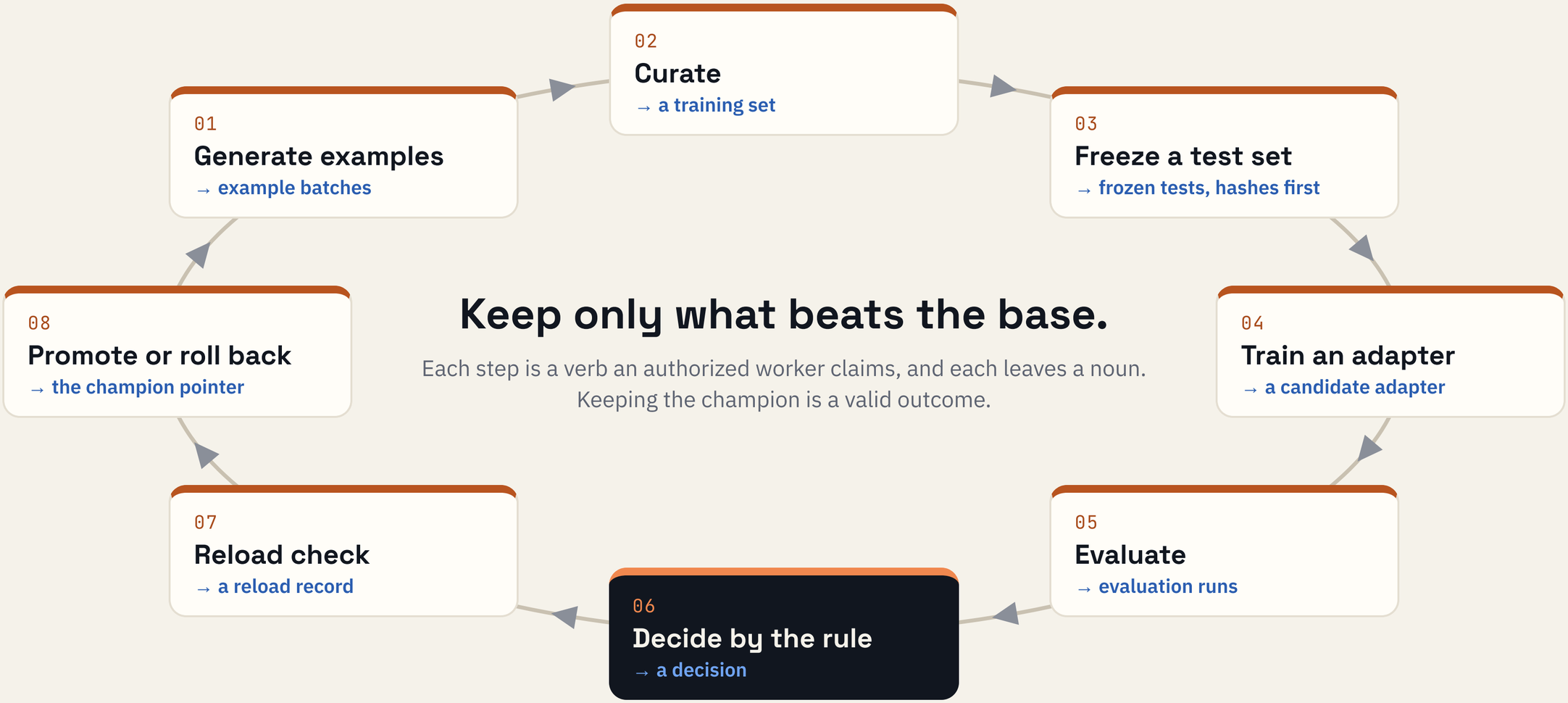
Unchanged, Muse Glimmer 30B answered 93.3% of a fresh tool-call test exactly right, and 96.7% of its replies were valid JSON.

THE FIX

Keep learning after deployment

Tune on checked examples of the job, and keep a tuned model only when it beats its base on a test it never saw, under a rule fixed in advance.

The loop


01
Generate examples
→ example batches
02
Curate
→ a training set
03
Freeze a test set
→ frozen tests, hashes first
04
Train an adapter
→ a candidate adapter
05
Evaluate
→ evaluation runs
06
Decide by the rule
→ a decision
07
Reload check
→ a reload record
08
Promote or roll back
→ the champion pointer
Keep only what beats the base.
Each step is a verb an authorized worker claims, and each leaves a noun.
Keeping the champion is a valid outcome.

Pre-registration and frozen tests

FIXED BEFORE ANY SCORE

FROZEN TEST
150 rows over tool families that training never sees, all pinned by one hash.

FRESH TEST
New rows each round, their hashes recorded before any model sees them.

THE PROMOTION RULE
Committed before the first training round, and pinned by a test.

NO LOOK-ALIKES
Fingerprints reject training rows that match a test row, its request or its template.

THE RULE SAID NO

Llama 3.2 1B, round 4 — Kept out
A new strategy lifted the development score from 61.25% to 76.25%, but the frozen test stayed at 50.7%.

Llama 3.2 3B, round 2 — Kept out
79.3% on the test, but schema validity fell 0.025, past its 0.02 limit.

Muse Glimmer 30B, 63 examples — Kept out
+5.3 points on the frozen test, but the 95% CI started at 0.0.

A guard written before any result stopped a real-looking gain.

The 1B and 3B results

LLAMA 3.2 1B

exact match by round

ROUND 0	ROUND 1	ROUND 2	ROUND 3	ROUND 4
12.0%	43.3%	43.3%	50.7%	50.7%
the unchanged base	promoted	kept the champion	promoted	kept the champion

LLAMA 3.2 3B

the same test and loop

ROUND 0	ROUND 1	ROUND 2	ROUND 3	ROUND 4
52.0%	72.7%	79.3%	not run	not run
the unchanged base	promoted	kept out by a guard		

63 owned examples removed the field failure: no typed value written as a string, and type correctness from 37.5% to 94.2% for the 1B.

Exact match on the frozen, synthetic 150-row test, scored strictly, with one seed per configuration.

The 30B result

MUSE GLIMMER 30B · QLORA ON FREE CLOUD GPUS

93.3% → 99.3%

Exact match on a fresh 150-row test whose hashes were registered before scoring. Trained on 511 synthetic rows.

+6.0 points
95% CI +2.0 to +10.7

10 / 1
rows better / worse

12 / 12
promotion-rule checks

8 / 8
rows matched on reload

The rule said promote. Nothing has been promoted or admitted, and the adapter weights aren't published.

Valid JSON rose from 96.7% to 100.0% and correct refusals from 97.3% to 99.3%, with 0 test rows matching the 511 training rows.

Learners: memory, paths and routing

Not all learning changes a model. Learner runs are ordinary verbs: they read execution reports and write their advice as versioned nouns.

MEMORY

Similar past work

Each run is fingerprinted. Before new work, recall finds similar past runs and how they went, failures included.

`memory · minhash-lsh`

PATHS

What succeeds

Runs are grouped by category, template, phrasing and shim into scorecards. They show associations, not causes.

`path_statistics · wilson-grouping`

ROUTING

Which way to try

Weights suggest a shim, method, harness or model, keep exploring the others, and move only with enough evidence.

`routing_weights · beta-bandit`

MODEL WEIGHTS

The third layer: tuning on real runs

It needs opted-in, verified, rights-cleared runs. That export is designed, not built yet.

Advice only: never a permission, never a hidden agent. The learners are implemented and tested, but not yet shown to improve real work.

Run it on free GPUs

FREE NOTEBOOK GPUS

A 30B model on two free T4s

Muse Glimmer 30B was tuned with QLoRA on Kaggle’s free notebook GPUs: 64 training steps in 83 minutes.

A GPU YOU OWN

1B and 3B on one RTX 3060

Every round of both on one 12 GB card: about 46 and 51 GPU-minutes, with no provider calls and no spending.

NO GPU

Check the plumbing first

A stand-in backend runs every verb, receipt, report and decision on a CPU. Its numbers describe the plumbing only.

```
# one tick plans and runs one round; every verb is claimed, kept alive and submitted for a receipt
$ PYTHONPATH=src:. python -m examples.continuous_tuning.cli tick \
   --root <runs> --base <base noun> --base-path <weights> --backend hf
```

The guide, its tests and the CPU backend: `tessera-infrastructure-work-and-storage-engine/docs/continuous-tuning.md`

Safety and rollback

A round that fails any check leaves the champion where it was.

FROZEN TESTS

Held-out tool families

Training never sees the test's tools, and every test set is pinned by its hash.

PRE-REGISTRATION

Hashes before scores

The rule and each fresh test are recorded before any model is scored on them.

CONTAMINATION CHECKS

No look-alikes

Fingerprints reject a training row that matches a test row, its request or its template.

RELOAD EQUIVALENCE

It must reload the same

The saved adapter, loaded onto a fresh copy of the base, must reproduce its replies exactly.

A RULE FIXED IN ADVANCE

Beat the base, break nothing

A gain whose lower bound is above zero, and no drop in valid output, refusals or general skills.

ROLLBACK

One pointer, every version kept

Adapters are never overwritten. Promotion moves one pointer, so rolling back moves it back.

Promotion changes no deployment, publication, admission or serving. Privacy is checked too: another key tried to read everything the 1B rounds wrote, and all 303 reads were denied.

Limits: what we don’t claim yet

- Everything is synthetic, with one task per result: typed tool calls for Tessera’s own tools.
- Each configuration ran once, with one seed, so nothing measures the spread between runs.
- The 150-row tests are small: they can miss a real gain of a few points.
- Nothing learns from live traffic yet. Rounds start on demand, and exporting real runs for training is designed, not built.
- Adapter weights aren’t published, and nothing has been promoted or admitted.
- The learners pass their tests, but haven’t yet been shown to improve real work.

Every kept-out round and every failure stays on the record.