

Sreyashi Chatterjee Adobe · AI Training Platform

Teaching the training platform **when a job is actually worth restarting** — and killing the ones that aren't.

1 The blind-restart trap

When a distributed job dies, torchrun's elastic agent restarts it — with **no idea whether it should**. A single bad assertion would loop through **4–5 blind hourly restarts**, burning GPUs, until a human finally noticed and killed it. A transient NCCL blip, meanwhile, deserves exactly that restart. Recovering **wrong**, at scale, is one of the largest sources of wasted training compute.

2 Signals we correlate

WHAT THE JOB EMITS
torch.distributed / NCCL · CUDA OOM · illegal memory access · SIGSEGV / SIGKILL · Python exceptions

WHAT THE NODE EMITS
gpud (ECC / NVLink / thermal) · NVIDIA DCGM diagnostics · Kubernetes Node Problem Detector

Every failure is classified by correlating both layers **across all ranks** — the way a human on-call would.

3 Failure taxonomy → recovery action

FAILURE CLASS	ACTION
Hardware fault	Cordon node + restart
Transient fault	Restart
Irrecoverable fault	Fail fast
Unknown no pattern matched	Diagnose & learn

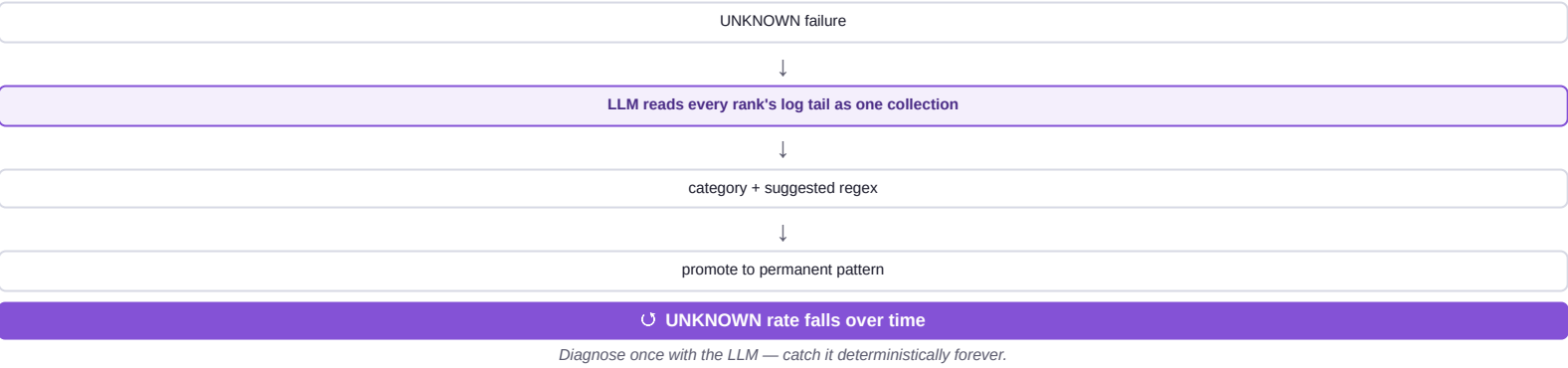
hardware fault — replace node

transient — retry

irrecoverable — stop now

unknown — learn a pattern

4 Closing the UNKNOWN gap



5 An LLM you can trust in a control loop

✓ Async, off the critical path

Runs only on UNKNOWNs and never blocks recovery.

✓ Derived, not self-reported confidence

Downgraded when the suggested regex matches no actual log line.

✓ Degrades safely to UNKNOWN

On low confidence or bad output — a wrong answer never triggers a wrong restart.

6 What's next: a calibrated classifier in the loop

✓ Classification, not generation

Route the category decision to a typed multiple-choice classifier instead of a free-text LLM call.

✓ Confidence comes for free

Calibrated, derived confidence per answer — reinforcing our existing trust-but-verify design.

✓ Reserve generation for the hard tail

Save the full LLM for regex synthesis only, cutting generative calls on the common path.

Impact

MEAN TIME TO RECOVER

Hours → ~3 min

Automated cordon / restart and fail-fast replace slow, manual recovery.

WASTED RESTARTS ON DOOMED JOBS

4–5 → 0

Irrecoverable failures killed on first detection, not looped through the restart budget.

FAILING JOBS STOPPED FASTER

~40%

of failing jobs now stopped on first detection instead of looping — freeing GPU-hours for useful work.