

Reward-Modulated Plasticity on a Frozen Production LLM

A persistent, reward-driven weight state kept **outside** the model: the 27B production backbone stays one shared, immutable copy. Adaptation with **no backprop**, **no optimizer**, **no retraining**, **no prompt-injected memory**. Every number below is measured.

WHY THIS MATTERS

WHERE FRONTIER COST SITS TODAY

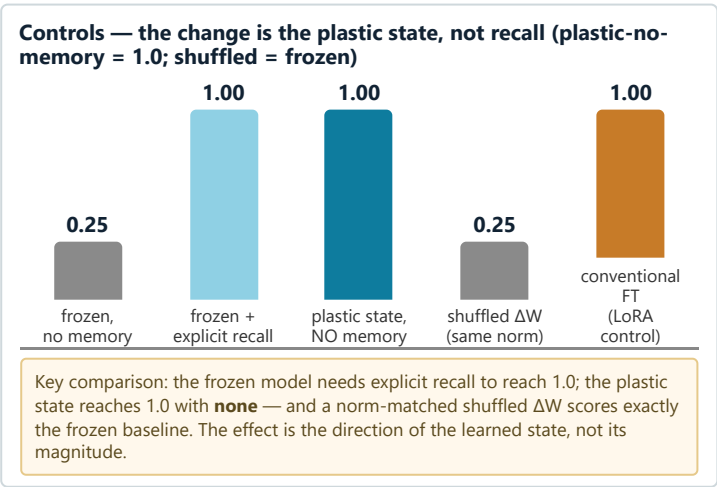
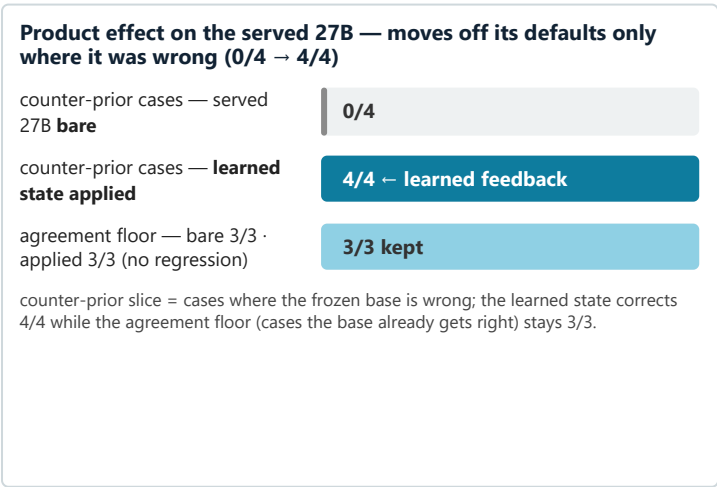
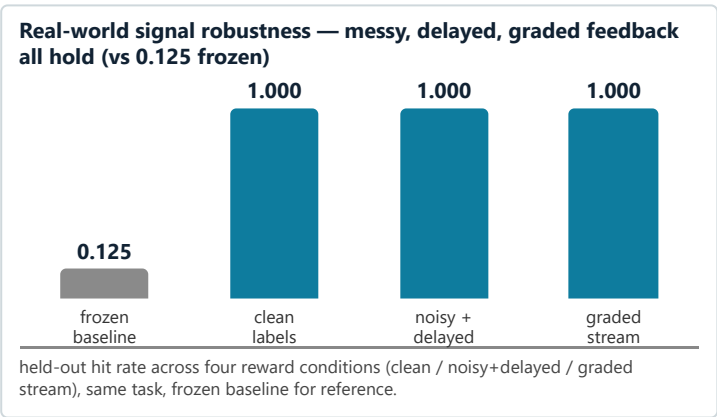
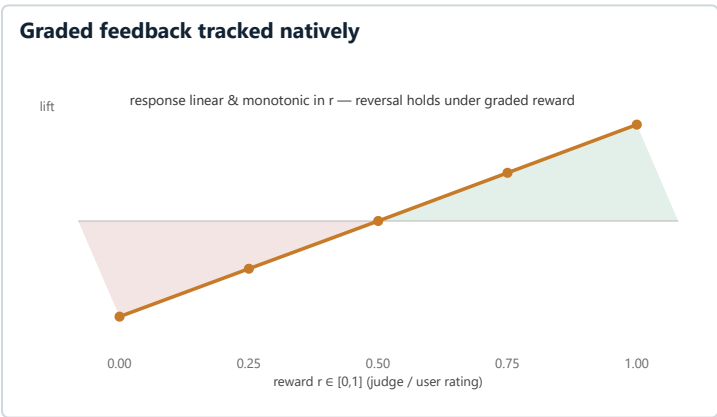
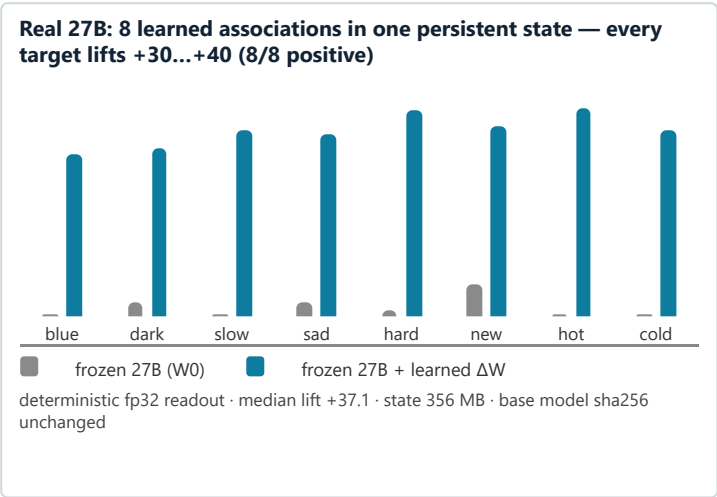
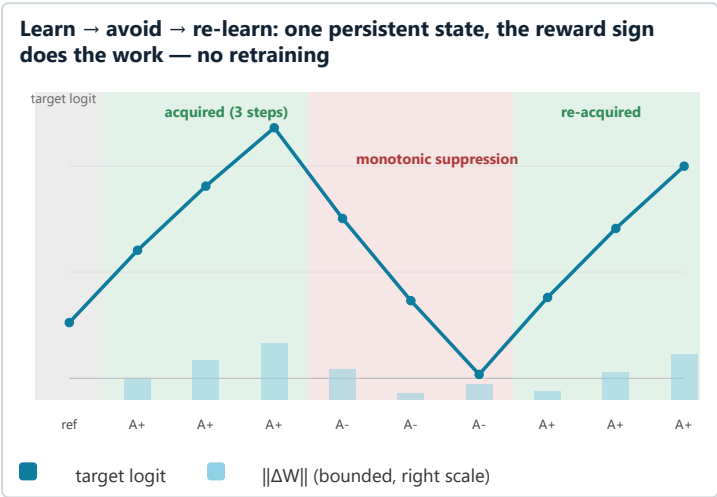
Frontier models are frozen after training; adaptation means iterative post-training. A published 2026 measurement (32B frontier-class model): reasoning post-training uses **17× the energy** of instruction post-training, ~87% of it in RL rollouts; the whole development process ≈ **12.3 GWh** and **15,887 kL water**. Every retrain — per task, per tenant — repeats that bill.

FRONTIER IS MOVING THIS WAY

2026 frontier research converges on the same problem: fast/slow weight learning for continual LLM adaptation; continual-learning-as-a-service for deployed agents; sovereign continual learning of open-weight frontier models; adapter hot-swapping. Common shape: **frozen base + a small, persistent per-task/per-agent adaptive state**.

WHAT WE HAVE = THE IMPACT

A **new learning / credit-assignment mechanism** at the extreme end of that line: local, reward-modulated plastic updates on a frozen 27B production model — no backprop, no optimizer, no replay, **zero training compute per adaptation**. Impact: per-agent adaptation collapses from GPU-hours to milliseconds + a 0.36 GB state; one frozen backbone serves many adaptive agents (60–130 measured in 47 GB RAM); VRAM freed for serving capacity.



Integrity proofs (every run): $\Delta W=0 \Rightarrow$ logits bitwise-identical to the untouched model · ΔW -only checkpoint → fresh process → behaviour reproduced bit-exact (max $|\Delta L| = 0.0$) with all conversation / memory / demonstrations removed · base model sha256 identical before/after every run · seed-reproducible, machine-readable per-update logs.

Real-World Advantage: the Cost of Adaptive AI, Rebuilt

Adaptive behaviour is normally bought with fine-tunes: backprop, optimizer state, GPU hours, and a model that drifts per tenant. This mechanism replaces that with a **tiny, persistent, reward-driven state per task/agent** — measured below on real hardware, including a production-scale probe on the real 27B.

ALIGNED TO FRONTIER DIRECTION — AND AHEAD OF IT

LESS TRAINING & ITERATING

Frontier iteration today = repeated post-training runs (measured 17× the energy of instruction tuning, ~87% in rollouts). A plastic state needs **zero training runs**: the update is one local reward-modulated step. Every avoided retrain removes that compute, carbon and water from the bill.

ACTUAL ADAPTATION AT DEPLOYMENT

Not a retrain, not a prompt — the model's behaviour changes from live reward and persists across restarts (measured bit-exact). That is genuine in-field adaptation the frozen base cannot do: a 4B + state reached hitA 1.0 where the frozen base scored 0.125, and it moved the served 27B off its wrong defaults 0/4 → 4/4.

BETTER HARDWARE UTILISATION

Learned states live in **host RAM, not GPU VRAM**; the PCIe link is idle during inference, so streaming them is ~free — **measured on the real 27B: the 357 MB state arrives in 13.2 ms, under one 21.6 ms decode round** (bit-exact). VRAM freed becomes KV headroom → more users / longer context per card. ~60–130 agents in 47 GB RAM; 15-domain bank designed.

What each adaptation actually stores — vs what fine-tuning must train

plastic state — 27B site

0.35 GB

plastic state — 4B site

0.10 GB

LoRA adapter (127× fewer params than ΔW)

0.79 MB

fine-tune weights (bf16, 27.8B)

55 GB

fine-tune optimizer state (Adam fp32, 27.8B)

222 GB

log scale. Plastic states: 27B site 0.35 GB / 4B site 0.10 GB · LoRA adapter 0.79 MB (fewer params than ΔW) · fine-tune of a 27.8B model: 55 GB weights *plus* 222 GB of Adam optimizer state — trained per tenant, per retrain.

One 32 GB card: shared frozen model + up to 4 active agents' state

KV headroom → more

frozen backbone (shared by all agents) — 24 GB

active agents' ΔW — 4 × 0.36 GB · the rest lives in **host RAM, not GPU**: ~60–130 adaptive agents in 47 GB

Each active agent's ΔW (0.36 GB) streams across the idle PCIe link — **measured on the real 27B: 13.2 ms < one 21.6 ms decode round**, hidden behind model compute, bit-exact, cold→hot swap verified. The remaining agents live in host RAM: **~60–130 adaptive agents in 47 GB**.

VRAM is the scarce resource in serving. Moving per-agent state out of VRAM converts it into KV headroom — more concurrent users and longer context on the same card at the same footprint.

Production-scale transfer — measured on the real 27B (stop-window probe, 2026-10-07)

ΔW block-stream, 1-site (357 MB)

13.2 ms

ΔW block-stream, 2-site (713 MB)

26.2 ms

one production decode round

21.6 ms

the ΔW block-stream is **shorter than one production decode round** → it hides behind a decode step at ~0 marginal wall-clock. Both 1-site and 2-site variants bit-exact vs resident; W0 sha256 unchanged. Artifact: hw/results/e_bank_step4_transfer.json.

The shared-state ceiling is escapable — per-domain banks (measured)

shared ΔW — concurrent domains recovered

0/4

per-domain banks (routed) — recovered

4/4

a single shared ΔW saturates the readout at ~1–2 concurrent behaviours (the measured capacity ceiling); the same events through a request→domain router + separate per-domain banks recover all 4/4. 15-domain bank designed for the 172-tool registry; prototype bit-exact, W0 unchanged. Artifact: harness/results/fork_b_bank_prototype_full.json.

Where it deploys — the same mechanism, validated end-to-end on CPU at zero GPU memory

EDGE / ROBOTICS

Smaller models (≤4B) with task-specific learned state on the device. Learns from local feedback (operator, sensor, outcome) — no cloud round-trip, no fine-tune farm. The state is one small file shipped with the hardware that keeps improving in the field: on-device adaptive AI.

MULTI-TENANT SAAS

One frozen backbone, per-tenant learned states in a RAM bank; each customer's preferences persist across restarts without any model forking. **Live in production today**: full capture → learn → deliver → measure pipeline, boot-persistent, rate-limited, audited; real user picks flow through the public route.

ON-PREM / REGULATED

The model never leaves the site — only a ~0.3 GB state per agent adapts. The base model is provably immutable (hash-verified), so the auditable artifact is the small learned state: inspect it, diff it, delete it. It consolidates to a rank-4 adapter (99.6% retention) for standard infra.

SMALLER MODELS FOR INDIVIDUAL TASKS

A 4B + learned state reached hitA 1.0 on adapted behaviour where the frozen base scored 0.125, and moved the served 27B off its wrong defaults 0/4 → 4/4. Right-sized models + adaptive state match much larger frozen models on the tasks they own.

Environmental & economic impact (inference from the measured anchors — scales with the number of avoided training runs)

Compute per adaptation

Zero training compute per request/tenant: one local reward-modulated step (<0.2 ms, no backprop, no optimizer) vs GPU-minutes-to-hours per fine-tune. Every avoided fine-tune removes that data-centre compute (energy + water) from the bill.

Hardware efficiency

Learned states in host RAM, not GPU VRAM; the PCIe link is idle during inference, so streaming them is ~free — **measured on the real 27B: 357 MB in 13.2 ms < one 21.6 ms production decode round** (bit-exact, cold→hot swap verified). VRAM freed → KV headroom → more users per card. ~60–130 agents on 47 GB RAM; 15-domain bank designed.

Longer model life

The shared base is provably immutable (hash-verified every run) and never drifts per tenant — one maintained model instead of N tenant forks. The state also **consolidates to a rank-4 adapter** (effective rank 1.16; rank-4 retention 99.6%, all 18 decision flips preserved) — it interoperates with standard serving infra.

Economics

Per-agent adaptation cost drops from training-run scale to a few-hundred-MB state file + milliseconds of update. Order-of-magnitude saving vs the incumbent per-segment fine-tune pipeline: **~\$0.1–1M/yr per enterprise (up to ~\$10M at heavy personalization)**; new markets: per-task device adaptation, per-tenant SaaS on one backbone, regulated on-prem.