Analog chips can do nonlinear ML — if you co-design around the device's native nonlinearity

W. Schulz

Jun 29, 2026

Abstract

On MNIST, a frozen MLP tolerates 4-bit weights with moderate device noise; 2-bit is the cliff. Saturating device nonlinearities match ReLU only after hardware-aware retraining (0.947 vs 0.938) — naive swaps fail. Class-IL routing stays robust at 4-bit with heavy noise. A first-order model, not a device simulation.

Deep-research synthesis, 2026-06-29. Two questions: (1) can analog neuromorphic chips do the nonlinear part of ML feasibly, and (2) what does it take to reverse-engineer ML algorithms to run on them? Grounded in current literature and reconciled against this repo's own photonic / wave / continual-learning results so we don't re-run what we already falsified.

The core asymmetry (the headline)

In photonics, the *linear* op (matmul via interference) is easy and the nonlinearity is hard — photons barely interact, so you need an exotic material or a detour through electronics. In analog electronics it is the exact opposite: the *linear* op is nearly free (Ohm's law sums currents, Kirchhoff adds them, a capacitor integrates — the physics does the matmul), and the nonlinearity is the awkward, costly part. Implementing a nonlinear activation in an analog crossbar "incurs significant time and energy penalties" and is usually pushed to the ADC periphery or to special devices (domain-wall, ferroelectric, synaptic/floating-gate transistors) [Nat. Commun. 2025].

So "can analog do nonlinear computation?" splits into two very different answers:

nonlinearity you want	feasibility
the device's native nonlinearity (transistor I-V, capacitor saturation, spike threshold, memristor dynamics)	free and natural — physics provides it
an arbitrary/programmable nonlinearity (a specific ReLU/GELU/softmax, precise & reproducible)	costly — needs ADC/digital periphery or careful pre-distortion; noisy and variable

Verdict on #1: nonlinear computation in analog is *feasible and even physically natural* — but only if you use the device's native nonlinearity and co-design the algorithm around it, rather than forcing an arbitrary digital activation. The moment you demand a precise, reproducible, arbitrary nonlinearity, you pay the ADC/precision tax that erodes analog's whole advantage.

What the literature establishes (cited)

- The matmul is the win; the nonlinearity is the tax. Analog/NVM crossbars deliver the matrix-multiply in O(1) via Ohm+Kirchhoff; the activation is the bottleneck and is typically done at the periphery (memristor-pre-distorted ramp ADCs) or via device physics [Nat. Commun. 2025 "nonlinear function approximation in analog crossbars"; AIP Appl. Phys. Rev. 2020].
- Accuracy survives — with hardware-aware training. ~89.6% CIFAR-10 (VGG8) at 5-bit under *real* memristor variations using a super-resolution strategy; 15× energy / 13× area wins. But naive mapping degrades badly — IR-drop, device-to-device variation, stuck-at-faults all cut accuracy [Nat. Mater. 2026; Sci. China Inf. Sci. 2024]. Co-design is not optional.
- ANN→SNN conversion works but isn't free. Quantization-/conversion-aware training recovers accuracy; naive topology mapping hits only 30–50% hardware utilization; hybrid ANN-SNN training is competitive [PMC 2023; Frontiers 2025].
- The field has converged on mixed-signal + CIM, precisely because pure-analog can't supply precise nonlinearity/memory cheaply [Nature Commun. 2025 commercialization review].

Reconciliation with this repo (what we already know)

- Our photonic cost model already found optical *compute* loses to electronic analog in-memory (AIMC) — so the analog substrate worth targeting for *computation* is AIMC (memristor/PCM/SRAM crossbars), not photonics. Photonics' durable role is *interconnect*.
- The dynamic-weight problem is the killer (our repeated finding): analog wins for static-weight, low-precision, always-on inference; it dies on energy/endurance when weights must change fast. So "reverse-engineer ML to analog" means inference-time mapping of a *frozen* trained net — not training on the chip.
- Our wave/complex substrate is decorative result means: don't bet on exotic wave-interference nonlinearity. Bet on the boring, real analog nonlinearity (device I-V, thresholds) and make the *algorithm* tolerate it.
- Our continual-learning result (forgetting = selection; ε-routing is the bottleneck and the lever, Expert-Gate router 0.62→0.935) gives a *specific* algorithm whose core op — sparse event-driven selection + in-place memory — is exactly what neuromorphic hardware does cheaply. That's the co-design match worth testing.

Reverse-engineering ML → analog: the falsifiable program

"Reverse-engineering" here = take a *trained* model and ask what survives analog execution, and what the algorithm must be re-shaped into. Pre-registered experiments (harness `neuromorphic/experiments/`, torch):

- E1 — ANN under analog non-idealities. A trained MLP run with weight quantization (4/6/8-bit), per-MAC additive read noise, and a device-shaped (not ReLU) nonlinearity. *Measure accuracy retention vs the digital baseline; locate the precision/noise cliff.* Kill: even 8-bit + tiny noise collapses → analog inference is hopeless without retraining.
- E2 — Nonlinearity substitution (the #1 question, made concrete). Replace ReLU with an analog-realistic *fixed, noisy* nonlinearity (saturating tanh-like / hard threshold / noisy device curve), with and without hardware-aware retraining. *Does the native nonlinearity compute as well as ReLU?* Kill: device nonlinearity can't match ReLU even after retraining.
- E3 — Does the CL selection primitive survive analog? Take our router solution (ε=0.055, 0.935) and add analog noise/quantization to the routing scores. *Does router ε stay low enough to still beat*

digital replay? This is the co-design payoff: the algorithm's bottleneck op run on the hardware that does it cheaply. Kill: analog noise explodes router ε.

The honest expectation, from the literature + our priors: E1/E2 pass with hardware-aware training, fail naive (co-design is mandatory); E3 is the genuinely novel test — if sparse selection is noise-robust, the analog-neuromorphic story for *continual* learning is real.

Sources

Nat. Commun. — nonlinear functions in analog crossbars · Nat. Mater. 2026 — high-accuracy memristor analog compute · Sci. China Inf. Sci. — non-ideality co-design · AIP Appl. Phys. Rev. — analog NVM architectures · PMC — ANN-to-SNN quantization-aware conversion · Frontiers 2025 — algorithm-hardware co-optimization · Nature Commun. — neuromorphic commercialization

---

Results (run 2026-06-29, torch, seeds 0/1, harness **`neuromorphic/experiments/`**)

E1 — trained ANN under analog non-idealities (MNIST MLP, digital baseline 0.938)

analog setting (naive map of a digitally-trained net)	ACC
8-bit weights	0.939
6-bit	0.939
4-bit	0.939
2-bit	0.186 (cliff)
6-bit + gnoise0.1/onoise0.05	0.936
6-bit + gnoise0.3/onoise0.2	0.912
6-bit + noise, hardware-aware retrain	0.941

The matmul side is robust. A frozen net runs naively at 4-bit + moderate device/read noise with almost no loss (≥0.93); 2-bit is the cliff; hardware-aware training holds accuracy under noise. Inference genuinely tolerates analog imprecision — consistent with the literature.

E2 — nonlinearity substitution (the #1 question): device-shaped vs ReLU (ReLU clean 0.938)

device nonlinearity	swap, no retrain	hardware-aware (with activation noise)
tanh (saturating)	0.691	0.932
satrelu (clamp 0–1, ReRAM-like)	0.897	0.947
thresh (binary spike)	0.248	0.739

Nonlinear computation in analog is feasible — if you co-design. Saturating device nonlinearities match or beat ReLU after hardware-aware retraining (tanh 0.932, satrelu 0.947 vs 0.938), even with 5% activation noise. Naive swap fails (co-design is mandatory). Binary spiking is the lossy extreme (0.739) — extreme activation quantization costs real accuracy. So "use the device's native nonlinearity and retrain around it" works; "force an arbitrary ReLU at inference" doesn't.

E3 — does the CL selection primitive survive analog? (digital router 0.935, replay bar 0.558) ★

analog setting	class-IL ACC	router ε
digital (fp32)	0.935	0.055
8-bit	0.934	0.056
6-bit + gnoise0.1/onoise0.05	0.936	0.054
4-bit + gnoise0.2/onoise0.1	0.927	0.063
4-bit + gnoise0.3/onoise0.2 (harsh)	0.832	0.158

The selection primitive is analog-robust (kill not fired). Even at 4-bit + heavy noise the reconstruction-router holds 0.832, far above the digital replay baseline (0.558); router ε only drifts 0.055→0.158. Why: routing is an *argmax over per-task scores* — you only need the right task's score to stay highest, not exact values, so noise must be large to flip the decision. Selection/routing is noise-tolerant by nature — which makes it an unusually good fit for noisy analog/neuromorphic hardware. This is the co-design payoff: the algorithm's bottleneck op (sparse selection) maps onto the hardware's strength *and* survives its non-idealities.

Synthesis — both questions, answered

- #1 Can analog neuromorphic do nonlinear computation feasibly? Yes, via the device's *native saturating nonlinearity* + hardware-aware retraining (E2: matches/beats ReLU). The matmul is the cheap, robust part (E1: 4-bit + noise naively fine); the nonlinearity is feasible but only when co-designed, not forced. Binary spiking is the lossy corner.
- #2 Reverse-engineering ML to analog neuromorphic? Inference-only mapping of a *frozen* net survives with hardware-aware training (E1 0.941); and the standout — the continual-learning selection primitive is analog-robust (E3 0.832 at 4-bit+heavy-noise > replay 0.558). The recipe: train digitally, compile to analog with the device nonlinearity in the loop, and lean on *selection* (which tolerates noise) rather than precise arithmetic.

Honest caveats

- MNIST flatters this — it's small and noise-tolerant; on harder data (CIFAR) the precision/ noise cliffs arrive sooner and the nonlinearity gap may widen.
- This is a first-order behavioral model (weight quantization, multiplicative conductance noise, additive read noise, fixed device nonlinearity, activation noise) — *not* a SPICE/device sim. It does not yet charge IR-drop, temporal drift, endurance, or ADC energy. "Survives" = survives the standard first-order non-idealities, not a full hardware validation.
- Inference-only / frozen weights — consistent with our repeated finding that dynamic weights are the killer for analog. On-chip *learning* is a different, harder question (see hard-CL track).

Published by NeuroGenesys (AW3 Technology, Inc.). Canonical version: https://neurogenesys.si/research/analog-nonlinearity-feasibility