LCNA

Persistent Local Learning and an Unexpected Retrieval Phenomenon in an Experimental Neural-Computing Architecture

Bounded development evidence from a local-learning architecture without global or local backpropagation

Shadi Dandashi · Independent Research · August 2026

ABSTRACT

LCNA investigates a different starting point for AI computation: rather than beginning with a conventional neural-network abstraction and changing how it is globally trained, it builds upward from individually stateful computational elements, local interaction, persistent state, and local learning. Its learning path contains no global backpropagation and no local backpropagation, and a frozen-runtime architecture verification found no optimizer-driven learning path in the audited runtime. Within a bounded reduced micronetwork, online persistent learning is demonstrated during active execution: under explicitly supervised teaching and contextual conditions, the Learned Cue moved from a response margin of 0 before learning to +12 after learning, identically across three development seeds, while an orthogonal comparison cue remained at 0, and the changed state remained available to later computation without requiring a separate backpropagation or optimizer-driven retraining stage. Causal dependence on the learned state is demonstrated by reset and inverse restoration (+12 → 0 → +12), and persistent writes were temporally gated in the tested conditions (250 ms: write observed; 1,500 ms and 6,000 ms: no comparable write). An unexpected retrieval phenomenon was then observed: a related unseen input and an older nominally unrelated control each expressed the learned response (+10) after learning, and causal localization showed that both accessed the same learned internal representation, whereas a prospectively designed route-disjoint control could not and remained at 0. However, the complete retrieval-specificity protocol did not qualify because two separate neutral-control requirements remained unmet in all three seeds, so the broader causal boundary is still being investigated. One controlled protocol currently takes more than a day on a consumer workstation, making experimental throughput the immediate limitation.

1 Introduction

Most contemporary deep-learning systems begin with a network abstraction trained using gradient-based optimization, typically through backpropagation [1, 2]. Many proposed alternatives modify the learning or credit-assignment procedure while retaining substantial parts of that abstraction [3, 4]. LCNA asks a different question. It investigates whether increasingly complex learning and computation can be built upward from *individually stateful computation, local interaction, persistent state, and local learning*, rather than beginning with a conventional neural-network abstraction and changing how that network is globally trained.

No global backpropagation. No local backpropagation. LCNA's learning path contains no gradient-based optimizer and no backpropagation of either kind. Learning arises through local state and local interaction.

It is important to be clear about what LCNA is *not*. It is not implemented as a conventional trainable neural-network abstraction, nor is it an image classifier or visual-only model. The current experiments use small visual inputs because vision offers a concrete and controllable sensory domain in which transformations, related inputs, specificity, retention, and later retrieval can be tested with matched controls. **Vision is the first experimental neural/sensory system, not the overall goal of LCNA.** The broader research direction concerns increasingly complex learning and computation, larger systems, richer inputs, and broader adaptive behaviour. None of those later stages is claimed here.

Sections 2–4 describe the architecture, the provenance-verified stimuli, and the methods; Section 5 presents results and their qualification status; Sections 6–11 cover rigor, runtime verification, compute, hardware, next questions, and the claim boundary.

2 Current experimental system

LCNA is an unconventional neural-computing architecture. The experiments reported here use a **bounded, reduced LCNA micronetwork**: a small experimental instance chosen so that every arm of the protocol can be matched and every comparison made exact. The architecture can be described by five properties (Fig. 1): each computational element carries its own state; elements interact only locally; learning is performed by those local state transitions and interactions; the resulting state persists after the learning experience ends; and signaling is spike/event-based rather than dense and synchronous, within the broader class of event-driven neural computation discussed in [5, 6].

The measured quantity throughout this document is a **bounded response margin** defined by the protocol. A margin of 0 is neutral; positive values report the designated learned response. The margin is a behavioural read-out of the micronetwork. This

communication reports measured behaviour and the controls that bound its interpretation.

Learning occurs within the runtime. In the current bounded prototype, controlled experience can modify persistent learned state during active execution. That changed state remains available to later computation without requiring a separate backpropagation or optimizer-driven retraining stage. The current experiments are explicitly supervised: controlled teaching and contextual signals provide the conditions under which local learning occurs.

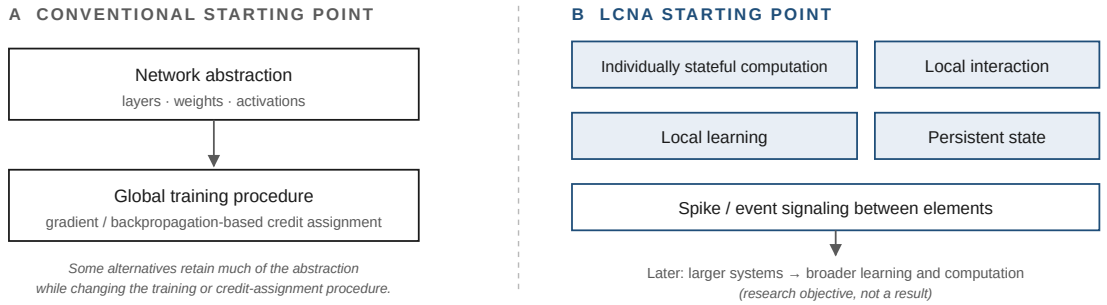

Figure 1. Computational starting point. (A) The conventional route begins with a network abstraction and applies a global training procedure. (B) LCNA begins with individually stateful elements that interact locally, learn locally, and retain persistent state, using spike/event signaling. The diagram is conceptual. The later stages named at the bottom are research objectives and have not been achieved.

3 Experimental stimuli

All results in this communication were measured with five visual stimuli (Fig. 2). Each is an exact experimental **6 × 6 binary raster**; the figures reproduce the frozen experimental rasters pixel-for-pixel using integer nearest-neighbour enlargement only, with no interpolation, cropping, rotation, recoloring, or annotation of the stimulus pixels. **Original experimental stimulus resolution: 6 × 6 binary pixels.** The resolution is deliberately low. The purpose of the current protocol is a controlled, bounded measurement of the persistent state that local learning produces, of its causal role in later behaviour, and of how specifically it is later expressed; small binary inputs keep every arm of the protocol matched and every comparison exact. Higher-resolution sensory learning is a later research stage (Section 10), not a property of the present system.

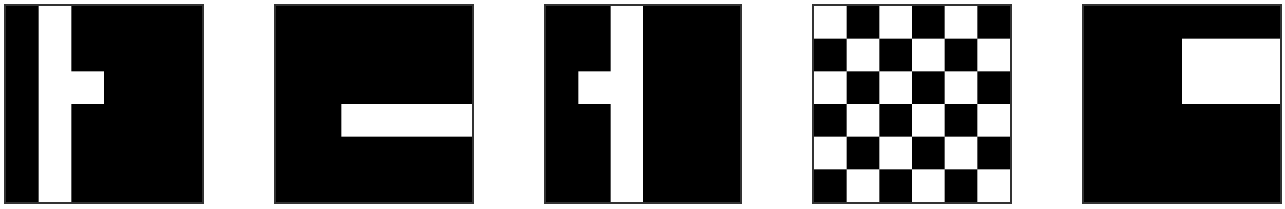

Figure 2. The five experimental stimuli. Each panel is the exact frozen 6 × 6 binary raster used by the protocol (0 = black, 1 = white, authoritative row order and orientation preserved), shown by integer nearest-neighbour enlargement. Roles are those defined by the protocol and verified in the public provenance record. No result in this document is inferred from the geometry of the rasters.

The five roles (Fig. 2) are defined by the protocol and verified in the public provenance record: the *Learned Cue* used during the controlled learning interval; an untrained *Orthogonal Comparison Cue* from which a neutral response is required; an untrained *Related Unseen Input* used as a transfer diagnostic; an *Older Nominally Unrelated Control*, a historical comparison retained as a known-overlap diagnostic and excluded from pass/fail after a later audit (Section 4.5); and a *Prospective Route-Disjoint Control*, a prospectively defined unrelated-specificity control from which a neutral response is required. Interpretations come from the audited protocol and its causal-localization results, not from the visual appearance of the rasters.

4 Methods

Methods are described at the level needed to evaluate the controls. All measurements below were made in the latest clean full development protocol: three development seeds; five pre-learning probes and ten post-learning probes per stimulus per seed; a frozen execution tree with checksummed atomic task receipts. No held-out final-qualification seed was executed.

4.1 Baseline

Every stimulus was probed before any learning took place in each seed, establishing a fresh pre-learning baseline. All five stimuli produced a margin of 0 at baseline. Later non-zero responses therefore cannot be explained by a pre-existing measured response; their relationship to the intervening learning experience is tested by the causal controls described below.

4.2 Local learning

Learning arises through LCNA's local stateful computation and local interactions, without global or local backpropagation, and it occurs within the runtime: in the experiments reported here, controlled experience produced persistent learned-state modification during active execution, and that state remained available to later computation without requiring a separate backpropagation or optimizer-driven retraining stage. The experiments are explicitly supervised. During the controlled learning interval the Learned Cue is presented together with a controlled teaching (instruction) signal and contextual conditions; the instruction signal is a protocol signal, not a visual stimulus. Cue-only, instruction-only, and causal-lesion control arms are summarized in Section 6.

4.3 Persistent-state causal test

After learning, the post-learning response to the Learned Cue was measured. The learned state was then reset from a checkpoint and the cue was probed again; finally the learned state was restored by inverse manipulation and the cue was probed a third time. The checkpoint round-trip was verified independently. This test asks whether the later response depends on the retained learned state, or merely coincides with having undergone the learning episode.

4.4 Temporal-credit test

Three preregistered timing conditions were tested across all three seeds: 250 ms, a 1,500 ms boundary condition, and a 6,000 ms beyond-window condition. The question is whether the component learning signals produce a persistent write only when they overlap sufficiently in time, or whenever they are merely present. **These are experimentally tested protocol conditions, not a claimed universal LCNA or biological time constant.**

4.5 Retrieval-specificity test

An earlier internal protocol had passed, but a fresh audit found that the Older Nominally Unrelated Control had acquired the learned response. Causal localization then showed that this control could access the same learned internal representation as the Learned Cue. The control was reclassified as a known-overlap diagnostic, and a new control was designed prospectively so that it could *not* access that representation. The successor protocol reported here probes the Learned Cue, the Orthogonal Comparison Cue, the Related Unseen Input, the older control, and the new Prospective Route-Disjoint Control before and after learning.

5 Results

ESTABLISHED BOUNDED RESULT

Online persistent learning — 0 → +12; causal reset / restoration +12 → 0 → +12; temporal gating of persistent writes. Demonstrated within the bounded development micronetwork.

SUPPORTED MECHANISTIC FINDING

Representation-mediated retrieval under the tested conditions — related unseen input +10 and older control +10 through causally localized access to the retained learned state; route-disjoint control 0.

REMAINING UNRESOLVED BOUNDARY

Neutral-control behaviour and state conversion — two neutral-control conditions changed when the protocol required them to remain absent, so the complete retrieval-specificity protocol did not qualify.

5.1 Persistent learning

The Learned Cue moved from a response margin of **0 before learning to +12 after learning**. The Orthogonal Comparison Cue remained at **0 → 0**. Both values were identical across the three development seeds and across the repeated probes within each seed (five pre-learning, ten post-learning). This establishes online persistent learning: a bounded learned state written during active execution that persisted beyond the learning episode and remained available to later computation (Fig. 3). The three development seeds are repeated development runs, not independent replications. The orthogonal cue's neutrality shows only that the designated response was not expressed for that comparison input.

5.2 Causal persistence: learned-state reset and inverse restoration

The learned-state manipulation produced the sequence **+12 → 0 → +12** in each development seed: the learned response was present after learning, absent after the learned state was reset, and restored after inverse restoration (Fig. 4). The reset checkpoint round-trip passed. Removing the learned state removed the response; restoring it restored the response. This is strong evidence that the later response depends on the retained learned state rather than on a transient output fluctuation or on an unrelated persistent state. The three conditions are three states of the *same* experiment on the same Learned Cue; they are not three different images.

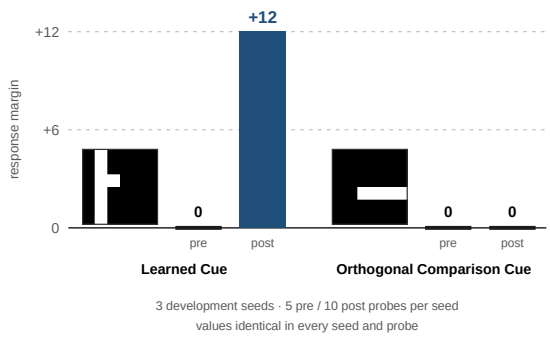

Figure 3. Persistent learning. Response margin before and after learning for the Learned Cue (0 → +12) and the Orthogonal Comparison Cue (0 → 0) in three development seeds, five pre-learning and ten post-learning probes per seed; values were identical across seeds and probes, so no error bars are shown. Supports a bounded persistent learned state with later behavioural consequence; does not establish general memory, semantic discrimination, or continual learning. BOUNDED RESULT

5.3 Temporal gating of the persistent write

A persistent modification was observed in the 250 ms condition in all three seeds, and no comparable persistent write was observed in the 1,500 ms boundary condition or the 6,000 ms beyond-window condition in any seed (Fig. 5). The presence of the component signals was therefore not sufficient; their timing mattered. **This supports temporal gating of the persistent learning-state write. It does not demonstrate recall of event time or temporal order.** The result concerns *when* persistent learning can be written, and only for the three tested protocol conditions.

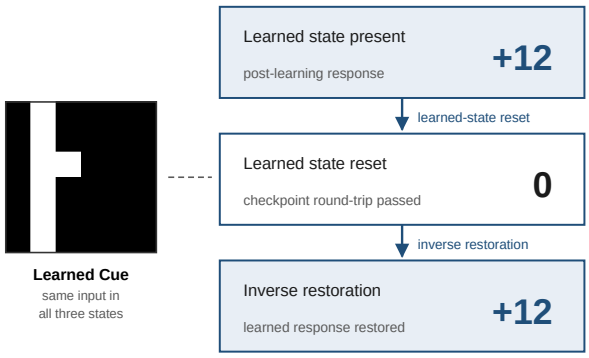

Figure 4. Causal dependence on the learned state. The same Learned Cue was probed with the learned state present (+12), after the learned state was reset (0), and after inverse restoration (+12), in each of three development seeds. These are three states of one experiment, not three images. Supports the conclusion that the post-learning response depends causally on retained learned state rather than on transient output drift; does not characterize what the state encodes. BOUNDED RESULT

5.4 Representation-mediated retrieval

The retrieval-specificity assay produced the response margins shown in Fig. 6, identically across the three seeds. The Learned Cue expressed the designated response (+12). Two inputs that had not been used in the learning association also expressed the learned response after learning: the Related Unseen Input (+10) and the Older Nominally Unrelated Control (+10). The learned cue, the related unseen input, and the older control remained distinguishable as complete input patterns; causal localization showed that the latter two could access the same learned internal representation. The Orthogonal Comparison Cue did not express the learned response (0). The Prospective Route-Disjoint Control, designed so that it could not access that representation, remained neutral (0), as predicted.

Tested timing condition	Persistent modification	Seeds
250 ms	■ Observed	3 of 3
1,500 ms boundary	☐ No comparable write	0 of 3
6,000 ms beyond window	☐ No comparable write	0 of 3

Figure 5. Temporal gating of the persistent write. Three preregistered timing conditions were tested in each of three development seeds; a persistent modification was written only in the 250 ms condition. Supports a controlled timing dependence of learning at write time. Does not establish a universal time constant and is not evidence of temporal-order or event-time memory. BOUNDED RESULT

5.5 Interpretation

The supported characterization of the combined results is **persistent storage + temporally gated learning + later representation-mediated retrieval**. In plain language: LCNA formed a persistent learned state. Timing affected whether that state was written. Later, distinct inputs that accessed the learned representation expressed the learned response, while the prospectively designed route-disjoint control did not. Representation-mediated retrieval was therefore causally localized to shared access to the learned representation. The current controls argue against explanations based solely on whole-pattern identity, late output coincidence, or a nonspecific response to every input. Which inputs can reach the representation was established by causal localization, not read off the rasters.

The remaining uncertainty is separate from the localized pathway. It concerns the broader retrieval-specificity boundary: two neutral-control conditions still changed when the protocol required them to remain neutral (Section 5.6). Until that boundary is resolved, the complete retrieval-specificity mechanism has not qualified.

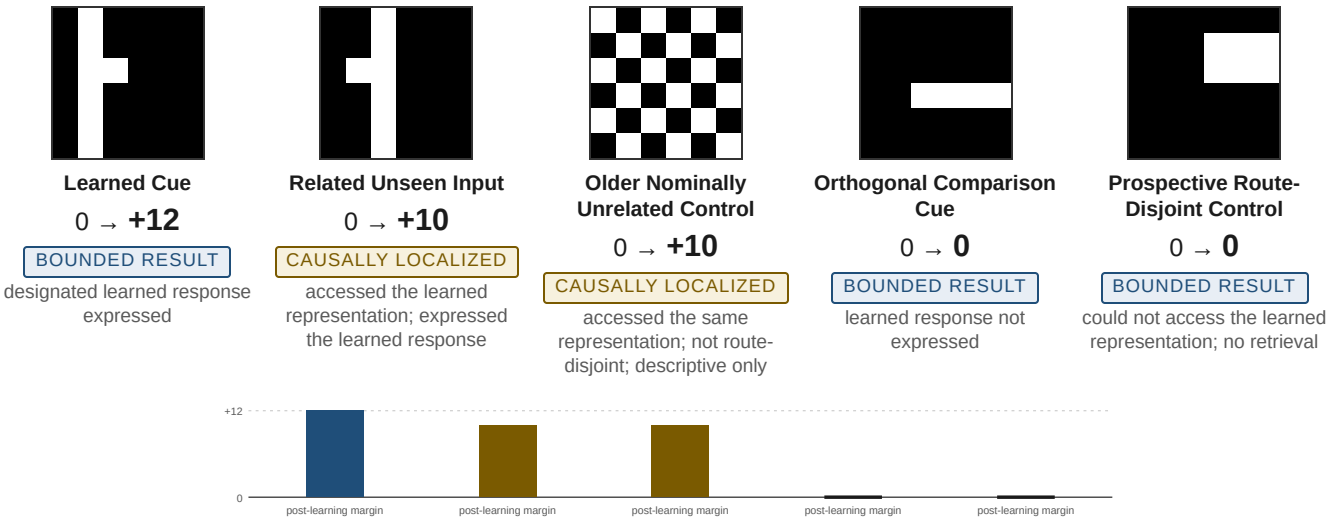

Figure 6. Stimulus-specific response margins after learning. Each panel shows the exact frozen 6 × 6 stimulus, its pre → post response margin (identical across three development seeds, ten post-learning probes per seed), and its evidence class; the bars plot the post-learning margin. This supports *persistent storage, temporally gated learning, and later representation-mediated retrieval*, with the retrieval pathway causally localized to shared access to the learned representation. **It is not evidence of semantic understanding or concept learning, and the broader retrieval-specificity boundary remains unresolved because separate neutral-control conditions failed (Section 5.6).** The older control was later found not to be route-disjoint; its result is descriptive and cannot establish unrelated-input specificity.

5.6 Qualification boundary: the latest retrieval-specificity protocol did not qualify overall

CURRENT QUALIFICATION STATUS

Although the persistent-learning, reset/restoration, timing, and prospective route-disjoint subresults remained supported, the latest complete retrieval-specificity qualification failed two separate neutral-control requirements. Two neutral-control conditions showed behavioural and internal-state changes that the protocol required to remain absent. Because both requirements failed in all three development seeds, the complete retrieval-specificity protocol did not qualify; the failure was reproduced in a clean full rerun. Consequently, persistent learning and its causal, temporal, and control-arm subresults remain supported bounded results; the representation-mediated retrieval pathway is causally localized; the broader retrieval-specificity boundary remains unresolved; and no held-out final scientific qualification has been performed. This outcome is reported deliberately: a favorable acquisition result and a passing prospective control do not, in this project, promote a behaviour to a qualified capability while a matched neutral requirement is unmet.

6 Experimental rigor

The methodological principle behind every protocol is that **a favorable output is treated as a hypothesis to attack, not as proof by itself.** If timing, pathway overlap, lingering state, checkpoint or scheduler behaviour, or a flawed control could explain a favorable output, that explanation is tested before the result is promoted. Table 2 lists the control classes in the current protocol and what each rules out. The cue-only, instruction-only, matched-time, and two causal-lesion arms all remained at margin 0 across the three seeds, so the learned response required more than exposure, instruction, elapsed time, or either isolated causal condition. Execution-integrity checks passed independently of the scientific outcome (Section 7).

Earlier apparent successes were rejected or reinterpreted when stronger audits exposed control overlap (the older control was found to share access to the learned representation, and the interpretation was revised from cue specificity to representation-mediated retrieval) and a defect in earlier development work. Results affected by those issues are not used as current scientific authority. The successor protocol introduced the prospective route-disjoint control, which passed, yet the protocol as a whole still failed its neutral-control requirements (Section 5.6) and is reported as such.

7 Frozen-runtime architecture verification

Status: verified with declared unknowns. A frozen-runtime architecture verification examined the current sealed execution path and directly tested the learning mechanism. Within the audited project path, no global or local backpropagation, autograd-based learning, optimizer-driven training, conventional trainable neural-network layers, fitted decoder, hidden trained-model pathway, or controller that directly manufactured the learned output was found. The learned-state change was positively traced into later computation: local learning modified persistent state, and later activity reused that retained state through the same computational pathway.

Table 2. Control classes in the current protocol.

Control class	Alternative explanation addressed	Outcome
Pre-learning baselines	Response existed before learning	0, all stimuli
Orthogonal cue	Response expressed for any input	0 → 0
Cue-only; instruction-only	Either component alone suffices	margin 0
Matched-time arm	Elapsed time alone suffices	margin 0
Causal lesions (two arms)	Response independent of the learning path's causal requirements	margin 0
State reset; inverse restoration	Transient drift or unrelated persistent state	+12 → 0 → +12
Temporal boundary; beyond-window	Signal presence, not timing, writes	write at 250 ms only
Prospective route-disjoint control	The learned response can appear without access to the learned representation	0 → 0
Checkpoint; scheduler equivalence	Result depends on scheduler or checkpoint	exact equivalence
Frozen execution; recovery integrity	Source drift, receipt reuse, or interruption misread as outcome	266 / 266 verified

Each control is described by the alternative explanation it removes.

The current experiments are supervised. Controlled cue, teaching/outcome, and contextual signals provide the conditions under which local learning occurs. The distinction the verification supports is *external teaching conditions + local learning state + persistent local modification*, rather than *loss + backpropagation + optimizer-driven parameter training*.

7.1 Declared limits

The current auditable runtime is sealed and reproducible under the bounded verification performed. The verification retains declared provenance and instrumentation limits: one historical wrapper identity remains unknown, and one fresh full-runtime instrumentation test was deferred under shared-machine resource-isolation constraints. These limitations do not provide evidence of a conventional learning pathway.

7.2 Evidence integrity

The result was preserved beyond a single output. *Checkpoint restoration:* learned state survived save/restore and returned the tested response. *Scheduler equivalence:* serial and parallel execution produced equivalent scientific state and trace results under the tested conditions. *Task receipts:* 266 checksummed task-completion receipts were preserved across the controlled protocol; they are task completions, not independent experiments or replications. *Runtime custody:* the current auditable runtime and associated evidence are sealed and checksummed.

8 Computational constraint

All experiments reported here ran on a single consumer workstation: an **AMD Ryzen 7 7700X**, an **NVIDIA RTX 4070**, and **32 GB RAM**. The latest clean full controlled protocol produced **266 checksummed task receipts** spanning approximately **27.36 hours** wall-clock from the first completed receipt to the last (Fig. 7). The 266 receipts are controlled task completions, not 266 independent scientific replications: they span pre-learning probes, experimental arms, post-learning probes, reset/restoration tests, temporal controls, and scheduler checks. The 27.36-hour figure is a wall-clock protocol window, not a benchmark asserting exclusive CPU or GPU utilization.

The bottleneck is therefore not one slow inference. A scientific interpretation requires matched controls, lesions or probes, challenge tests, and reruns, so the wall-clock cost of one question is multiplied across many runs. Each cycle of the workflow in Fig. 7 currently takes more than a day, and a single protocol revision (for example, adding the prospective route-disjoint control) repeats the full cycle. Rigorous qualification multiplies experimental runtime, and that multiplication now limits the rate at which the open questions in Section 10 can be answered.

9 Hardware mismatch and the FPGA research direction

9.1 Workload structure versus present hardware

LCNA's computation includes many individually stateful elements that interact locally and signal by spikes/events. GPUs are optimized for regular, massively parallel arithmetic on dense vector and matrix workloads with throughput-oriented SIMD/SIMT execution [7]. CPUs are flexible general-purpose processors but are not physically organized around massive distributed local-state and event processing. The defensible formulation is:

> **LCNA is presently being emulated on CPU/GPU hardware optimized around materially different workload structures.**

This is a **motivated hardware hypothesis** grounded in the character of the computation, not a demonstrated efficiency result. It does not assert that GPUs cannot execute spiking systems, that event signaling automatically reduces energy consumption, or that LCNA is already more efficient than anything.

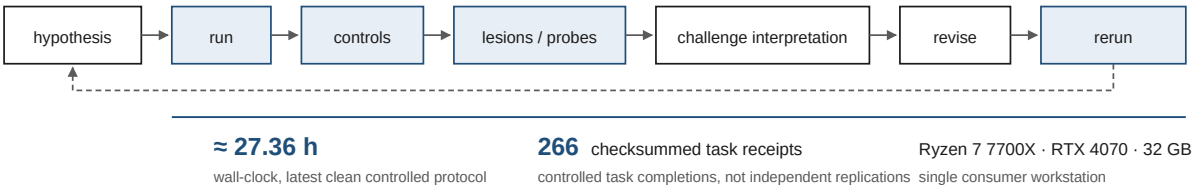

Figure 7. The experimental iteration cycle and its current cost. Shaded stages are the ones the workstation executes; the others are analysis and design. One complete clean controlled protocol produced 266 checksummed task receipts over approximately 27.36 hours wall-clock. This supports the claim that qualification, not a single inference, is the runtime bottleneck. It is not a CPU- or GPU-only benchmark and does not support a historical speed trend across earlier, differently configured experiments.

9.2 Two reasons for FPGA research

Potential acceleration. Reconfigurable hardware may execute LCNA's event-driven, local-state computation more efficiently than the present software-emulation path, shortening the iteration cycle in Fig. 7. This has not yet been demonstrated, and no speed-up figure is claimed.

Computational-architecture exploration. Because an FPGA is reconfigurable [8], it permits experimental variation of high-level hardware organization (state placement, event routing, scheduling, local execution, memory organization, and communication) while preserving the causal computation under study. The central research question is: **what hardware organization best matches LCNA's computation?** The objective is to accelerate the actual LCNA computation, not to replace it with a cheaper approximation. No hardware design is proposed here; that is an outcome of the research, not a premise of it.

10 Current status and next questions

Current. Bounded online persistent learning with causal reset/restoration and temporal write gating, established in a reduced micronetwork without global or local backpropagation; and a representation-mediated retrieval phenomenon whose pathway is causally localized but whose broader specificity boundary is unresolved (Section 5.6).

Immediate. Localize why the neutral arms still changed and determine which causal condition is missing from the current explanation; resolve the current causal boundary of the retrieval behaviour; and design the successor controlled protocol with prospectively independent controls and a clean held-out qualification stage.

Next scale. A larger visual/sensory learning architecture for more demanding transformation, specificity, representation, and retention experiments (multiple learned states, interference, longer retention, development of internal representations), while preserving falsifiable controls.

Longer-term research. Test how far the same local, stateful foundation can extend to larger systems, richer inputs, multiple learned states, longer retention, and increasingly varied adaptive behaviour. These are research questions, not promised outcomes; each will require new evidence and new qualification gates.

> **The immediate practical limitation on answering these questions is experimental throughput.** A single controlled protocol on the present workstation takes more than a day; every hypothesis, control revision, and rerun pays that cost again.

11 What the current evidence supports

SUPPORTED

- online persistent learning during active execution
- causal use of retained learned state (reset / restoration)
- temporal gating of learning at write time
- bounded representation-mediated retrieval under the tested conditions
- an audited learning path without backpropagation or optimizer-driven training, under explicit experimental supervision

NOT ESTABLISHED

- unsupervised or lifelong continual learning; unrestricted autonomous adaptation
- semantic understanding or concept learning
- temporal-order or episodic recall
- superior scaling, computational efficiency, or energy efficiency; FPGA acceleration
- literature-wide architectural novelty

No held-out final qualification of the retrieval behaviour has been performed.

12 References

1. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986).
2. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
3. Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J. & Hinton, G. Backpropagation and the brain. *Nature Reviews Neuroscience* **21**, 335–346 (2020).
4. Bellec, G., Scherr, F., Subramoney, A., Hajek, E., Salaj, D., Legenstein, R. & Maass, W. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature Communications* **11**, 3625 (2020).
5. Maass, W. Networks of spiking neurons: the third generation of neural network models. *Neural Networks* **10**, 1659–1671 (1997).
6. Roy, K., Jaiswal, A. & Panda, P. Towards spike-based machine intelligence with neuromorphic computing. *Nature* **575**, 607–617 (2019).
7. Nickolls, J. & Dally, W. J. The GPU computing era. *IEEE Micro* **30**(2), 56–69 (2010).
8. Kuon, I., Tessier, R. & Rose, J. FPGA architecture: survey and challenges. *Foundations and Trends in Electronic Design Automation* **2**(2), 135–253 (2008).

Evidence and provenance. All LCNA results in this communication are taken from the LCNA public technical evidence package (evidence snapshot 25 August 2026) and its stimulus–result map; the references above provide background only and are not evidence for LCNA's results. The five displayed stimuli are verified exact public exports of the frozen experimental rasters. Their experimental roles and result mappings were verified against the authoritative experiment evidence; image digests and custody details are retained in the separate public provenance record.

Scope. This communication reports experimental behaviour and the controls that bound its interpretation.