Brain-CA: A Dual-Speed, Low-Energy Architecture for Cognitive Learning via Cellular Automata

Jerry Felix
Brain-CA Technologies, Inc.
Sarasota, FL
jfelix@brain-ca.com

Steve Brunker
Brain-CA Technologies, Inc.
Cincinnati, OH
sbrunker@brain-ca.com

Carol Hibbard
Brain-CA Technologies, Inc.
Ocala, FL
chibbard@brain-ca.com

ABSTRACT

Brain-CA addresses the growing energy demands of AI—especially in data centers reliant on GPU-driven, brute-force neural networks. It offers a low-power alternative: a novel architecture built on binary modeling, dual-speed processing, and Cellular Automata. By separating physical structure from logical function, Brain-CA enables distributed cognition and efficient learning through wave interaction and fast path-based prediction. With lightweight Estimators, bifurcated logic, and minimal computation, the system solves tasks like XOR efficiently. This paper outlines the architecture and supporting biological and computational validation.

CCS Concepts

• **Computer systems organization → Architectures → Parallel architectures → Cellular architectures** • Computing methodologies → Machine learning → Machine learning algorithms • Theory of computation → Models of computation → Probabilistic computation

Keywords

Cellular Automata; low-power AI; distributed learning; binary modeling; dual-speed processing; energy-efficient computation; wave-based communication; Estimator; Cincinnati Algorithm.

1. INTRODUCTION

Today's AI systems rely on massive neural networks trained on GPUs—powerful but inefficient machines that simulate learning through brute-force weight updates. Brain-CA takes a different path, redefining learning at the binary level and building intelligence from the bottom up.

Using a distributed network of identical hexagonal cells, Brain-CA enables low-energy cognition through wave interaction, fast path prediction, and lightweight model updates. This paper presents its core innovations: physical/logical separation, dual-speed architecture, and Estimators powered by the Cincinnati Algorithm. Results from software simulations will be shared, along with a roadmap toward hardware implementation in both edge and data center contexts.

2. ARCHITECTURAL OVERVIEW

2.1 Physical Architecture

Brain-CA's physical layer is a uniform grid of identical hexagonal cells. Each interacts locally with six neighbors—there is no central control. Intelligence emerges from their simple, coordinated interactions [8, 9, 10].

2.2 Logical Architecture

Each cell participates in three overlapping subsystems:

- **Communication** spreads waves to discover correlations.
- **Memory** stores correlation patterns from wave collisions.
- **Connection** builds fast prediction paths.

These subsystems use a mix of persistent memory and transient state for continuous, distributed operation.

3. DUAL-SPEED PROCESSING

Brain-CA uses two speeds to support learning and inference [2]:

Low-Speed Layer

Wavefronts radiate from stimuli cells outward. Collisions between wavefronts are used to detect correlations. In broadcast mode, waves persist briefly, allowing the system to detect timing-based relationships—similar to biological temporal windows [5]. In directed mode, the Connection Subsystem routes messages neighbor-to-neighbor to build or remove fast communication paths.

High-Speed Layer

Once a path is in place, a stimulus can immediately trigger a prediction. Responses may be delivered instantly or routed spatially to arrive at the right time.

This dual-speed design mirrors biology, where slow sensing transitions into fast reaction [4].

4. ESTIMATORS AND BINARY MODELING

At the core of Brain-CA learning is the Estimator—a lightweight model that captures the ratio of binary outcomes using a simple, low-energy update process called the Cincinnati Algorithm [3].

Setup

Before learning or prediction, each storage bit is paired with a random bit. The Estimator then identifies:

- The leftmost *mismatch* between storage and random bits.
- The rightmost match to the *mismatched* storage bit.

This quick setup yields all the information needed for the next step.

Learning
When an observation arrives:

- If no *mismatch* was found, the model grows by appending the observed bit. This growth is rare and probabilistically gated, similar to Morris' Approximate Counting [7].
- If a *mismatch* was found and the observation matches the *mismatched* random bit, all bits from the rightmost match position onward are inverted. Otherwise, no change is made.

This allows the model to adapt efficiently, especially during early learning.

Prediction
Estimators support two modes:

- **Best Guess** returns the high-order bit—the dominant historical outcome.
- **Probabilistic Simulation** uses the storage bit at the *mismatched* position. If no *mismatch* was found, the Estimator can retry with a new random pairing or flip a coin.

These modes mirror how humans either commit to a decision or consider multiple outcomes.

Each Estimator is fast, compact, and consumes minimal energy, making it ideal for large-scale, distributed cognition.

5. THE ROLE OF RANDOMNESS

Brain-CA uses randomness to fairly regulate learning and prediction. Random bit pairings ensure early updates are frequent, but taper off as confidence grows. This mechanism aligns with findings from Columbia University [1], which show that randomness plays a critical role in unbiased memory formation in biological systems.

6. COLLISION AND CORRELATION TRACKING

When waves from two stimuli collide, the intersecting cell tracks correlations using up to four Estimators: B's value when A is true, B's value when A is false, A's value when B is true, and A's value when B is false. Typically, only the first two are useful for forward prediction when A precedes B, while the others are often discarded. These compact models enable efficient pairwise learning throughout the grid.

7. CONNECTION SUBSYSTEM AND FAST PREDICTION

When a sufficiently significant correlation is found, the Connection Subsystem builds a fast path using directed neighbor-to-neighbor signals. Each cell along the route closes a switch, linking the input source to the predictive cell. This enables immediate or sequenced predictions and complements the exploratory wavefronts—shifting the system from discovery to real-time response.

8. BIFURCATION: BEYOND PAIRWISE RELATIONSHIPS

To capture nonlinear patterns like XOR, Brain-CA uses bifurcation—splitting models based on the condition of another signal (e.g., "B when A is true" vs. "B when A is false"). This allows complex logic to emerge naturally, without layers, weights, or thresholds.

9. BIOLOGICAL AND EVOLUTIONARY PARALLELS

Brain-CA mirrors key features of biological intelligence: wave-like signal propagation resembles brain waves; memory emerges from temporal co-occurrence; and predictive pathways echo synaptic strengthening. Most importantly, its low-energy design aligns with evolutionary pressures favoring efficient cognition [6].

10. CONCLUSION

Brain-CA represents a shift in AI architecture—replacing brute-force learning with lightweight, binary modeling. Through wave-based discovery, fast path prediction, and distributed bifurcation, it enables energy-efficient cognition.

Its math-free, modular design supports both edge devices and data centers. In the cloud, it can replace power-hungry GPU clusters; at the edge, it delivers real-time intelligence where GPUs cannot go.

This paper introduces the core architecture. We will share simulation results at the workshop, along with plans for hardware deployment and scaling.

11. REFERENCES

[1] Columbia University. 2024. *Randomness and Memory Formation in Biological Systems*. *Proceedings of the National Academy of Sciences (PNAS)*. https://doi.org/10.1073/pnas.2316745122

[2] Jerry Felix, Steve Brunker, Carol Hibbard. 2024. *Casting off the Old Guard: Achieving Superior A.I. Performance through Simplification* https://brain-ca.com/wp-content/uploads/2024/06/Casting-off-the-Old-Guard.pdf.

[3] Jerry Felix, Steve Brunker, Carol Hibbard. 2024. *The BRAIN-CA™ Estimator and The Cincinnati Algorithm: Simplification Tools for Artificial Intelligence*. https://brain-ca.com/wp-content/uploads/2024/06/The-BRAIN-CA™-Estimator-and-The-Cincinnati-Algorithm.pdf.

[4] Harvard University. 2023. *Wave-Based Communication in Neural Tissue*. *Nature*. https://doi.org/10.1038/s41586-023-06147-2

[5] Eugene M. Izhikevich and Gerald M. Edelman. 2008. *Large-scale model of mammalian thalamocortical systems*. *PNAS* 105, 9 (2008), 3593–3598.

[6] Simon B. Laughlin, Rob R. de Ruyter van Steveninck, and John C. Anderson. 1998. *The metabolic cost of neural information*. *Nature Neuroscience* 1, 1 (1998), 36–41.

[7] Robert Morris. 1978. *Counting Large Numbers of Events in Small Registers*. *Communications of the ACM* 21, 10 (1978), 840–842.

[8] Brain-CA Technologies. 2024. *Artificial intelligence based on cellular automata*. U.S. Patent No. 11,847,386 B1.

[9] John von Neumann. 1951. *The General and Logical Theory of Automata*. In *Cerebral Mechanisms in Behavior*, L.A. Jeffress (Ed.), 1–41.

[10] Stephen Wolfram. 2002. *A New Kind of Science*. Wolfram Media.

12. ADDENDUM: Addressing Reviewer Feedback

We thank the reviewers for their insightful and constructive feedback. The following responses address the key areas of inquiry, clarification, and comparison raised during review:

1. Applications and Use Cases

Q: *Could Brain-CA eventually replace today's brute-force neural network computation?*
A: Yes, Brain-CA is designed as a general-purpose learning and inference system and can eventually serve as a substitute for certain classes of neural network applications—particularly where energy efficiency, real-time responsiveness, or edge deployment is critical. Tasks such as **image classification**, **diagnostics**, **sensor fusion**, and **temporal sequence prediction** are especially well-suited, as Brain-CA's event-driven, binary models thrive in environments where relationships between input streams carry more weight than raw pixel-by-pixel matching.

That said, Brain-CA is not a direct drop-in for dense-matrix CNN pipelines; instead, it redefines learning as the **emergence of predictive relationships from discrete events**. In areas like autonomous navigation, manufacturing defect detection, or conversational interfaces where relationships and timing matter, it can deliver results with dramatically lower power and infrastructure requirements.

2. How Brain-CA Solves Tasks Like XOR

Q: *How does Brain-CA solve tasks like XOR?*
A: The XOR logic function cannot be solved by linear models because it requires understanding **conditional dependence**. Brain-CA addresses this via **bifurcation**: it builds two separate Estimators—one for "B when A is true" and one for "B when A is false." These allow the system to model non-linear relationships **without hidden layers or backpropagation**.
For instance:

- If A=1, it consults Estimator$_1$ to predict B.
- If A=0, it consults Estimator$_2$.

  By selecting the appropriate Estimator based on the conditioning signal, XOR is solved using only simple storage and logic operations.

3. Clarifying the Architecture with Examples

Q: *Could you provide clearer illustrations or examples of how Brain-CA works?*
A: We agree this is essential. A detailed visual example will be presented at the workshop showing:

- A small hexagonal cell array with three data sources: A, B, and C.
- Ripple propagation from simultaneous stimuli.
- A collision cell at the midpoint of A and B capturing "B when A is true" using Estimators.
- Later, when A is observed, a fast path prediction is routed to predict B.

This step-by-step animation will help attendees **see how relationships are learned, bifurcated, and then used for fast inference**.

4. Encoding in Brain-CA

Q: *How is data encoded for Brain-CA, given that encoding is a challenge in AI?*
A: Unlike CNNs that rely on convolution filters to find visual features, Brain-CA accepts **pre-encoded binary streams** as input. In practice, this allows flexibility:

- For images, bits may represent individual pixel intensities, segments, or high-level extracted features.
- For sound or time-series data, binary transitions represent events or thresholds.
- For sensor arrays, bits can correspond to sampled digital states.

The system **does not require a fixed encoding strategy** and instead learns to associate patterns based on **spatial-temporal correlations**. Over time, better encoding strategies can evolve to maximize collision efficiency or Estimator utility, which we are exploring in ongoing MNIST-based benchmarks.

---

5. Comparisons to Other Brain-Inspired Models

Q: *How does Brain-CA compare with hyperdimensional computing or spiking neural networks (SNNs)?*
A:

Architecture	**Energy Efficiency**	**Hardware Complexity**	**Learning Method**	**Temporal Support**
Brain-CA	Extremely high (bit-level ops, no math)	Low (identical hex cells)	Direct, probabilistic	Built-in dual-speed, ripple timing
SNNs	Moderate to high	Moderate	Spike-timing-dependent	Yes
Hyperdimensional	Moderate	High (vector ops)	Symbolic/random projection	Indirect

Brain-CA stands out in its **fundamental simplicity**—no matrix multiplication, no real-valued weights, no gradient descent. Its hardware is inherently suited for bitwise operations and **causal learning**. Unlike hyperdimensional systems that manipulate dense vectors, or SNNs that still rely on spikes as approximated neurons, Brain-CA starts from binary events and builds **logic-like predictions** through **real-time environmental interaction**.

6. Scalability and Limitations

Q: *Can Brain-CA handle large datasets or compete with modern LLMs?*
A: Brain-CA does not aim to replicate LLMs word-for-word but rather to handle large-scale **causal inference tasks** with vastly less compute. Its learning mechanism scales logarithmically with experience due to probabilistic gating, and inference remains lightweight due to fast path activation.

The current prototype can model tens of thousands of Estimators in parallel and grows dynamically. Hardware scaling (currently underway) will unlock massively parallel training on real-world data streams. That said, generative capabilities are limited to probabilistic sampling of past patterns—not abstract reasoning or language synthesis (at least not yet).

7. Hardware Support Plans

Q: *What are your hardware plans?*
A: Our software simulations already run on multi-threaded CPU environments. We are currently migrating to **FPGAs hosted on AWS**, which allow us to test bit-level logic and routing behaviors efficiently. The next phase includes:

- Bit-true FPGA implementation of Estimators.
- Cellular routing and collision logic across thousands of cells.
- A roadmap for eventual **ASIC deployment**, where the uniform hex-cell structure can be etched directly onto silicon for edge inference chips.

These steps will be covered in our workshop presentation, including **performance projections and energy savings** vs. neural net baselines.