

Emergent Computation

Emergent Computation

An

Analysis of the Physical Computation Paradigm Shift from Abstract Logic to Embodied Dynamics

Author: Rowan Brad Quni-Gudzinis **Affiliation:** QNFO **Contact:** rowan.quni@qnfo.org **ORCID:** 0009-0002-4317-5604 **ISNI:** 0000 0005 2645 6062 **DOI:** <https://papers.qnfo.org/papers/emergent-computation/> **Publication Date:** 2025-09-30 **Version:** 1.0

The classical Turing-von Neumann architecture, which has defined the digital epoch, is confronting insurmountable energetic and structural limits, most notably the von Neumann bottleneck. This necessitates a paradigm shift toward physical computation, a framework where information processing is understood not as an abstract sequence of logical operations but as an emergent property of the intrinsic dynamics of physical systems. This report articulates the principles of this new paradigm, which replaces direct, top-down control with a model of indirect influence: a complex physical system is guided by minimal perturbations, and its resulting high-dimensional state is interpreted by a trained observer.

The canonical architecture for this approach is identified as Physical Reservoir Computing (PRC), which leverages the rich, non-linear dynamics of a fixed physical substrate (the “reservoir”) to perform complex feature extraction, with only a simple linear “readout” layer requiring training. A comprehensive survey of potential computational substrates is presented, including biological neural networks, dynamically stabilized quantum systems, and engineered electromagnetic metamaterials, demonstrating the universality of the PRC framework. The implications of this paradigm are transformative, pointing toward a new science of “computational matter,” a programming model based on inverse design and objective function architecture, and a revised epistemology where concepts like time and randomness are understood through the lens of computational irreducibility and the bounded observer. While significant challenges in substrate engineering, theoretical mapping (the isomorphism problem), and reliability remain, this report provides a comprehensive roadmap for the dawning epoch of embodied computation, where the distinction between the computer and the physical world dissolves.

****1.0 Introduction: The Physical Computation Paradigm**

Shift** |

****1.1 The End of an Epoch: The Foundational Limits of the**

Turing-von Neumann Architecture** |

The computational discipline stands at a precipice, facing the culmination of an era defined by the Turing-von Neumann architectural model. For decades, this abstract, logic-driven framework has delivered exponential growth, but it now confronts fundamental physical limits that signal the end of its epoch. The von Neumann bottleneck, a structural consequence of separating memory and processing, imposes an insurmountable energy and latency ceiling (Backus, 1978). At the quantum level, Landauer’s principle establishes a fundamental thermodynamic limit to computation, dictating that each irreversible bit operation must dissipate a minimum quantum of energy, a constraint that silicon-based systems are now approaching (Landauer, 1961). This convergence of limitations necessitates a fundamental reconception of computation itself, moving beyond the abstraction of logic gates and towards a new epoch where computation is recognized as an emergent property of physical dynamics. |

****1.2 The Dawn of a New Epoch: Computation as an Emergent**

Property of Physical Dynamics** |

A transformative perspective is emerging that reconceptualizes computation as an inherent phenomenon arising from the natural dynamics of physical systems when properly configured and interpreted. This paradigm shift dissolves the artificial boundary between “hardware” and “software” by recognizing that all computation ultimately occurs within physical substrates governed by natural laws. This perspective acknowledges that biological systems, such as the human brain, achieve extraordinary computational efficiency not by approximating digital logic but by leveraging the intrinsic dynamics of their physical components (Mead, 1990). Modern research in reservoir computing demonstrates how complex physical systems, from networks of neurons to electromagnetic fields, can naturally process information through their high-dimensional state space dynamics when appropriately perturbed and observed (Jaeger, 2004; Tanaka et al., 2019). This approach transforms the engineering challenge from building abstract logical gates to discovering, shaping, and interpreting the natural computational properties of physical matter. |

****2.0 The Foundational Inefficiencies of Classical**

Computation** |

The staggering energy inefficiency of conventional computing represents not merely an engineering challenge but a fundamental consequence of the architectural choices that define the Turing-von Neumann paradigm. When compared against biological computation, silicon-based systems exhibit energy expenditures that are typically five to eight orders of magnitude higher for comparable computational tasks. This energetic chasm stems from the deliberate discarding of analog information through binary quantization, the energy-intensive process of maintaining precise voltage levels against thermal noise, and the massive overhead associated with data movement between physically separated components. |

****2.1 The Energetic Chasm: Orders of Magnitude Disparity**

Between Silicon and Biological Systems** |

The human brain performs complex cognitive tasks with an energy budget of approximately 20 watts, while supercomputers tackling comparable problems consume megawatts of power. This disparity, exceeding five orders of magnitude, represents one of the most profound inefficiencies in modern engineering. This chasm arises because biological systems achieve efficiency not through miniaturization but through fundamentally different computational principles. Neurons operate with analog dynamics, exploit the natural physics of ion diffusion, and integrate memory and processing within the same physical structures (Mead, 1990). Conventional computing, by contrast, expends enormous energy maintaining precise binary states against thermal noise and moving data across architectural boundaries that biology never created. |

****2.2 The Architectural Constraint: The Von Neumann**

Bottleneck and the Artificial Hardware/Software Dichotomy** |

The von Neumann bottleneck—the structural limitation imposed by the separation of processing and memory units—represents not merely a performance constraint but a fundamental architectural flaw that becomes increasingly dominant as computational demands scale (Backus, 1978). In modern systems, the energy cost of moving a single bit of data from memory to a processor can exceed the energy required to perform hundreds of arithmetic operations. This architectural separation creates an artificial dichotomy between “hardware” and “software” that forces all computation to be expressed through a narrow instruction pipeline, deliberately discarding the rich analog

dynamics of the physical substrate in favor of a simplified binary abstraction. The resulting framework requires complex software layers to simulate parallelism and concurrency that physical systems naturally exhibit, introducing massive overhead that biological systems avoid. |

****3.0 Principles of the New Paradigm: Physics as the**

Ultimate Substrate** |

The emerging physical computation paradigm rests on a profound reconceptualization: computation is not something we impose upon matter through abstract logical operations, but rather a natural phenomenon that emerges from the dynamics of physical systems when properly configured and interpreted. This perspective dissolves the artificial boundary between “computation” and “physics,” recognizing that all information processing ultimately occurs within physical substrates governed by natural laws. The unifying principle across this spectrum is that computation arises from controlled physical dynamics, where the computational process corresponds to state space traversal and the computational output manifests as stable attractor states. |

****3.1 The Spectrum of Physical Abstraction: From Digital**

Logic to Substrate Isomorphism** |

The relationship between computational processes and their physical substrates exists along a continuous spectrum of abstraction. |

****3.1.1 High Abstraction: The Universal but Inefficient**

Digital Model** |

Conventional digital computing represents the extreme of high physical abstraction, deliberately constructing an artificial computational environment that is effectively divorced from the underlying physical substrate. This approach achieves remarkable universality and reproducibility by forcing physical systems to approximate idealized Boolean logic. However, this universality comes at a tremendous cost in energy and efficiency, as the system discards the rich analog dynamics of the physical substrate, treating these dynamics as noise to be suppressed rather than resources to be exploited. |

****3.1.2 Low Abstraction: The Efficient but Specialized**

Embodied Algorithm** |

At the opposite end of the spectrum lies the embodied algorithm approach, where computation emerges directly from the natural dynamics of a physical system specifically configured to solve a particular class of problems with extraordinary efficiency. These systems exploit the inherent non-linear dynamics, resonance properties, and state space structure of their physical substrates to perform computation through their natural evolution. This efficiency, however, comes at the cost of flexibility, as each embodied algorithm is typically specialized for a particular class of problems. |

****3.2 The Unifying Principle: Computation as Controlled**

Physical Dynamics** |

The physical computation paradigm rests on a unifying principle: computation is fundamentally the controlled evolution of a physical system’s state space. Inputs correspond to perturbations of the system’s dynamics, processing occurs through the natural evolution of the system’s state, and outputs emerge as stable attractor states

that can be interpreted as solutions. This principle applies universally across computational substrates, from silicon transistors to neural networks to quantum systems. |

*****3.2.1 State Space Traversal as the Computational***

Process** |

In the physical computation paradigm, the computational process is the trajectory a physical system follows through its high-dimensional state space. Unlike conventional computing, which decomposes problems into sequential binary operations, this approach leverages the system's natural evolution through its possible configurations to embody the calculation. The richness of this approach lies in the dimensionality of the state space; complex non-linear systems possess state spaces with dimensionality far exceeding what could be efficiently represented in conventional architectures, allowing them to process information in massively parallel ways. |

*****3.2.2 Attractor States as the Computational***

Output** |

The computational output in physical systems emerges not as a deliberately constructed result but as stable attractor states toward which the system naturally evolves. These attractor states exist within the system's high-dimensional state space and correspond to local minima in the system's energy landscape. In biological systems, these manifest as stable patterns of neural activity that represent perceptions or decisions; in reservoir computing systems, they correspond to stable configurations of the reservoir's state that can be linearly decoded to produce the desired output. |

*****3.3 The Operational Mode: Resonance, Perturbation, and***

Interpretation** |

The operational framework of physical computation centers on three interrelated principles that replace conventional notions of programming and processing: resonance, minimal perturbation, and interpretive observation. |

*****3.3.1 Resonance and High-Q States as Energy-Efficient***

Computational Modes** |

Resonance phenomena provide the foundation for energy-efficient computation in physical systems by enabling selective, high-gain responses to specific inputs while maintaining operation near thermodynamic equilibrium. High-Q (high quality factor) resonant states allow physical systems to store energy with minimal dissipation, creating computational modes where information processing occurs through the selective amplification of specific frequencies or patterns while suppressing irrelevant noise. In neural systems, for example, gamma-band oscillations (30 — 100 Hz) serve as high-Q resonant states that enable selective communication between neural populations with minimal metabolic cost. |

*****3.3.2 Minimal Perturbation as the Programming***

Input** |

Programming in the physical computation paradigm shifts from constructing explicit instruction sequences to applying minimal, carefully crafted perturbations that initiate specific state space trajectories within the physical substrate. This approach recognizes that the computational power resides primarily in the substrate's natural

dynamics rather than in the input signal itself. In biological systems, this manifests as sparse neural coding; in reservoir computing implementations, it corresponds to low-dimensional input projections that perturb a high-dimensional physical reservoir just enough to initiate complex, solution-containing dynamics. |

****3.3.3 Interpretive Observation as the Processing**

Core** |

The computational process in physical systems culminates not in the explicit construction of results but in interpretive observation, where the final state is actively interpreted through a trained mechanism that translates complex physical configurations into meaningful outputs. This approach recognizes that the physical substrate's state space contains vastly more information than can be directly extracted, requiring a trained interpreter to select and decode the relevant patterns. In reservoir computing systems, for example, the readout layer is typically a simple linear model trained to map the reservoir's high-dimensional state to the desired output. |

****4.0 The Canonical Architecture: Physical Reservoir**

Computing (PRC)** |

Physical Reservoir Computing (PRC) represents the canonical implementation of the physical computation paradigm, providing a structured framework that harnesses the natural dynamics of physical systems for efficient computation (Tanaka et al., 2019; Nakajima et al., 2021). This architecture transforms complex physical systems into powerful computational substrates by leveraging their high-dimensional state space dynamics through a three-component structure: an encoder, a reservoir, and a readout. The genius of PRC lies in its separation of concerns: the physical reservoir's complex dynamics are fixed, requiring no training, while computational flexibility is achieved through a simple, trainable readout mechanism that learns to interpret the reservoir's states. |

****4.1 Conceptual Framework of the PRC**

Architecture** |

The PRC architecture implements the physical computation paradigm through a structured three-component framework. The encoder translates abstract problems into physical perturbations appropriate for the specific substrate. The reservoir itself constitutes the computational heart of the system—a fixed, complex physical system whose rich, non-linear dynamics perform the core computational work by transforming input perturbations into high-dimensional state representations. The readout mechanism completes the architecture by interpreting the reservoir's complex states and translating them into meaningful outputs, typically implemented as a simple, trainable model that learns to recognize solution patterns within the reservoir's state space. |

****4.1.1 The Encoder: Translating Abstract Problems into**

Physical Perturbations** |

The encoder serves as the critical interface between abstract computational problems and the physical reservoir, transforming high-level inputs into precisely calibrated perturbations that initiate the appropriate computational trajectories. In biological implementations, this corresponds to sensory processing pathways; in photonic reservoirs, it involves modulating light intensity or phase; and in superconducting implementations, it requires precise current or flux injections. |

****4.1.2 The Reservoir: The Physical Substrate as a**

High-Dimensional Feature Extractor** |

The reservoir is the computational heart of the PRC architecture—a fixed, complex physical system whose rich, non-linear dynamics naturally transform low-dimensional input perturbations into high-dimensional representations. This component leverages the inherent complexity of physical systems to create computational power through dimensionality expansion. The reservoir’s effectiveness depends on key dynamical properties: non-linearity, fading memory, and a high-dimensional state space. |

****4.1.3 The Readout: The Trained Linear Interpreter of**

Complex System States** |

The readout mechanism completes the PRC architecture by interpreting the reservoir’s complex high-dimensional states and translating them into meaningful computational outputs. This component embodies the principle of interpretive observation as the processing core. The readout is typically a single-layer perceptron or linear regression model trained to map the reservoir’s state vectors to the desired outputs, with the training process focusing exclusively on this interpretation layer. |

****4.2 The Training Phase: Supervised Learning as the**

Bridge to Interpretation** |

The training phase in PRC shifts the computational burden from modifying complex physical systems to training simple interpretive mechanisms. Rather than adjusting the reservoir’s internal parameters, training focuses exclusively on the readout mechanism, teaching it to recognize which patterns within the reservoir’s high-dimensional state space correspond to meaningful solutions. This process begins by exposing the reservoir to a set of known inputs while recording its resulting state trajectories. The training algorithm then solves a relatively simple optimization problem: finding the linear transformation that best maps these recorded reservoir states to the desired outputs, typically through standard techniques like ridge regression. This approach avoids the computationally intensive backpropagation through time required by conventional recurrent networks, instead leveraging the reservoir’s natural dynamics to handle temporal dependencies and non-linear transformations. For a formal derivation of the optimal readout weights, see Appendix A. |

****5.0 A Survey of Potential Computational Substrates**

(Reservoirs)** |

The physical computation paradigm reveals that virtually any complex physical system with appropriate dynamical properties can serve as a computational substrate. These substrates share key characteristics that enable effective reservoir computing: rich non-linear dynamics, fading memory, and sufficient dimensionality. The diversity of viable substrates underscores the universality of the physical computation principle—that computation is not something we impose upon matter but a natural phenomenon that emerges from appropriately configured physical dynamics. |

****5.1 The Biological Reservoir: The Brain as a Resonant,**

Plastic Substrate** |

Biological neural systems represent nature’s most sophisticated implementation of the physical computation paradigm. The brain achieves extraordinary computational efficiency by leveraging the intrinsic dynamics of its physical components (Mead, 1990). Central to this efficiency are neural oscillations, which serve as high-Q resonant

states enabling selective communication. Complementing these resonant dynamics is neuroplasticity—the brain’s intrinsic mechanism for adapting its structure and function—which serves as a natural implementation of the readout training process. |

****5.2 The Quantum Reservoir: Dynamically Stabilized**

Coherent Systems** |

Quantum systems offer a uniquely powerful substrate for physical computation by leveraging superposition and entanglement to access state spaces of extraordinary dimensionality (Ghosh et al., 2022). Quantum Reservoir Computing (QRC) uses dynamically stabilized coherent systems as high-dimensional computational resources where information is processed through the evolution of quantum states. While maintaining quantum coherence presents significant challenges, dynamically stabilized systems—where quantum states are actively maintained through feedback or periodic driving—show promise for creating practical quantum reservoirs. While their adaptation to reservoir computing remains speculative, ultra-low-power superconducting logic devices such as Quantum Flux Parametrons, which use the polarity of a quantum magnetic flux to represent information, have potential for high-Q resonant states that could be harnessed in such a context. |

****5.3 The Electromagnetic Reservoir: Harnessing Fields**

Through Interrogation and Interpretation** |

Electromagnetic fields in engineered materials provide a versatile substrate for physical computation, where information processing occurs through the interrogation and interpretation of field dynamics. Computational metasurfaces, in particular, represent engineered electromagnetic structures designed with specific non-linear properties that enable real-time signal processing through their interaction with incident waves. We propose the Interrogative-Interpretive Computing (IIC) model as a generalization of reservoir principles to such substrates, where computation is a two-stage process: interrogative procedures that probe the physical system with specific inputs, and interpretive analysis that extracts meaningful information from the system’s response. This approach, grounded in the interpretive observation principles of quantum reservoir systems, transforms materials science into computational engineering by creating “computational matter” whose physical properties are engineered to solve specific problems. |

****5.4 The Complex Systems Reservoir: Emergent Dynamics in**

Large-Scale Networks** |

Large-scale complex networks—from power grids to financial markets—exhibit emergent computational properties through their collective dynamics. Power grids, for example, demonstrate remarkable computational capabilities through their collective response to disturbances, where the propagation of frequency deviations across the network effectively performs distributed optimization to maintain stability. The computational power of these complex systems arises from thermodynamic information compression—the process by which high-dimensional inputs are transformed into lower-dimensional outputs through the system’s natural dynamics. |

****6.0 Implications of the Physical Computation**

Paradigm** |

The physical computation paradigm represents not merely a technological evolution but a profound reconceptualization with far-reaching implications across engineering, programming, and epistemology. This paradigm shift necessitates a transformation in engineering practice from designing abstract logical circuits to

creating “computational matter” with specific, tunable non-linear dynamics. The programming model undergoes an equally profound transformation, shifting from constructing explicit instruction sequences to defining objective functions that guide the evolution of physical systems. |

****6.1 The Engineering Shift: The Science of “Computational**

Matter”** |

The physical computation paradigm necessitates a fundamental transformation in engineering practice from designing abstract logical circuits to creating “computational matter” with specific, tunable non-linear dynamics. This new discipline focuses on designing materials whose physical properties are engineered to solve specific classes of problems through their natural response to external stimuli, effectively embedding computation within the material itself. This involves integrating sensor and actuator arrays at the substrate level, creating materials that can both perceive their environment and respond computationally without requiring external processing units. |

****6.2 The Programming Shift: From Logic Design to**

Objective Function Architecture** |

The physical computation paradigm necessitates a profound transformation in programming practice—from constructing explicit instruction sequences to defining objective functions that guide the evolution of physical systems toward solution-containing states. This new paradigm, objective function architecture, recognizes that the programmer’s role is shifting from specifying every step of the computation to designing the conditions under which the system will naturally evolve toward solutions. This approach embodies the principle of inverse design: starting with the desired computational function and evolving the physical form that will naturally implement it. |

****6.3 The Epistemological Shift: The Role of the**

Computationally Bounded Observer** |

The physical computation paradigm forces a profound reconsideration of epistemological foundations by recognizing that all computation occurs within physical systems subject to thermodynamic constraints. This perspective reveals that **computational irreducibility**—the principle that some systems’ behavior cannot be predicted without effectively simulating each step of their evolution—is a fundamental property of physical systems (Wolfram, 2002). Computational irreducibility provides a physical basis for our experience of time and causality. Furthermore, this perspective reframes longstanding philosophical questions about determinism, randomness, and free will within a computational framework where apparent randomness may arise from computational irreducibility rather than true indeterminism. |

****7.0 Critical Challenges and Open Research**

Questions** |

Despite the transformative potential of the physical computation paradigm, significant challenges remain. These challenges are not merely technical but conceptual, requiring new theoretical frameworks that integrate dynamical systems theory, thermodynamics, and computational theory into a unified understanding of physical computation. |

****7.1 The Isomorphism Problem: The Search for a Universal**

“Compiler”** |

The isomorphism problem represents perhaps the most fundamental theoretical challenge in physical computation: the lack of a universal framework for mapping arbitrary computational problems onto physical substrates with different dynamical properties. Unlike conventional computing, where the universal Turing machine provides a theoretical foundation, physical computation lacks an equivalent framework because the computational efficiency depends critically on how well the problem structure aligns with the substrate's natural dynamics. |

****7.2 The Substrate Engineering Problem: From Theoretical**

Models to Physical Fabrication** |

The substrate engineering problem represents a formidable practical challenge: translating theoretical models of computational matter into physically realizable materials with precisely engineered non-linear dynamics, memory properties, and resonance characteristics at the required scales. This requires advances in nanofabrication, materials science, and multi-scale modeling that are still in their infancy. |

****7.3 The Training and Repeatability Problem: Managing**

Chaos, Cost, and Reliability** |

The training and repeatability problem presents a critical practical challenge for PRC systems: achieving reliable training and consistent operation despite the inherent variability and chaotic nature of many physical substrates. Physical reservoirs, particularly those operating near the edge of chaos where computational power is maximized, are inherently sensitive to initial conditions and parameter variations, making consistent training difficult. |

****8.0 Conclusion: The Dawn of the Embodied Computation**

Epoch** |

****8.1 Synthesis of the Paradigm Shift from Abstract Logic**

to Embodied Dynamics** |

The physical computation paradigm represents a synthesis of insights spanning physics, neuroscience, and computer science that fundamentally reimagines computation not as an abstract sequence of logical operations but as an emergent property of physical dynamics. This synthesis dissolves the artificial boundary between “hardware” and “software” by recognizing that all computation ultimately occurs within physical substrates governed by natural laws. The resulting framework achieves extraordinary energy efficiency by operating near thermodynamic equilibrium, leveraging resonance and minimal perturbation, while maintaining flexibility through the separation of the physical reservoir's fixed dynamics from the trainable readout mechanism. |

****8.2 A Roadmap for the Development of Physical Reservoir**

Computers and Computational Matter** |

The path forward for realizing the transformative potential of physical computation requires a coordinated research agenda that addresses the critical challenges while building on current successes. In the immediate term, research should focus on developing standardized benchmarking frameworks for PRC systems. In the medium term, the field must address the isomorphism problem by developing mathematical frameworks for mapping computational problems onto physical substrates. In the long term, the integration of physical computation principles into mainstream engineering practice will require the development of computational matter design tools that enable inverse design. The ultimate goal is not merely more efficient computers but a fundamental integration of computation with physics that transforms how we understand and interact with the material world. |

Appendix A: Formal

Derivations

This appendix provides a rigorous mathematical foundation for the core principles of physical computation discussed in this report.

Mathematical

Foundations of Physical Computation

Axiom 1 (Computational State Space) Every physical computational system $\mathcal{S}$ is characterized by a state space $\Omega \subseteq \mathbb{R}^n$ where each state vector $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_n(t)]^\top$ represents the complete dynamical configuration of the system at time t .

Axiom 2 (Physical Dynamics) The temporal evolution of $\mathcal{S}$ is governed by a generally non-linear dynamical system:

$$\frac{d\mathbf{x}}{dt} = \mathbf{F}(\mathbf{x}, \mathbf{u}, t) + \xi(t) \quad (A.1)$$

where $\mathbf{u}(t) \in \mathbb{R}^m$ is the input perturbation, $\mathbf{F} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^n$ describes the system's intrinsic dynamics, and $\xi(t)$ represents stochastic noise.

Definition 1 (Physical Reservoir Computer) A Physical Reservoir Computer (PRC) is a triple $\mathcal{R} = (\mathcal{E}, \mathcal{S}, \mathcal{O})$ where: - $\mathcal{E} : \mathcal{I} \rightarrow \mathbb{R}^m$ is the encoder mapping input space $\mathcal{I}$ to physical perturbations - $\mathcal{S}$ is the physical substrate obeying Axiom 2 - $\mathcal{O} : \mathbb{R}^n \rightarrow \mathcal{Y}$ is the readout function mapping reservoir states to output space $\mathcal{Y}$

Theorem 1 (Von Neumann Energy Bound) For classical von Neumann architecture, the minimal energy per computation satisfies:

$$E_{\text{von Neumann}} \geq k_B T \cdot \mathcal{C}(\{\mathbf{i}_k\}, \{\mathbf{y}_k\}) + \alpha \cdot N \cdot d_{\text{memory-processor}} \quad (A.2)$$

where $\mathcal{C}$ measures computational complexity, N counts memory accesses, and d is the memory-processor distance.

Proof: 1. By Landauer's principle: $E_{\text{Landauer}} \geq k_B T \ln 2$ per bit erased. 2. For $\mathcal{C}$ -complex computation: $E_{\text{comp}} \geq k_B T \cdot \mathcal{C}$. 3. Data movement energy: $E_{\text{movement}} = \alpha \cdot N \cdot d$. 4. Thus: $E_{\text{total}} \geq E_{\text{comp}} + E_{\text{movement}} = k_B T \cdot \mathcal{C} + \alpha N d$.

Definition 2 (Substrate Isomorphism) A computational substrate $\mathcal{S}$ is isomorphic to problem $\mathcal{P}$ if there exists a diffeomorphism $\phi : \Omega_{\mathcal{P}} \rightarrow \Omega_{\mathcal{S}}$ such that:

$$\phi \circ \mathbf{F}_{\mathcal{P}} = \mathbf{F}_{\mathcal{S}} \circ \phi$$

where $\mathbf{F}_{\mathcal{P}}$ and $\mathbf{F}_{\mathcal{S}}$ are the natural dynamics of the problem and substrate respectively.

Theorem 2 (Reservoir State Evolution) Given input sequence $\{\mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_T\} \subset \mathcal{I}$, the reservoir state evolves as:

$$\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + \int_{t_k}^{t_{k+1}} \mathbf{F}(\mathbf{x}(\tau), \mathcal{E}(\mathbf{i}_k), \tau) d\tau + \xi_k$$

where ξ_k represents integrated noise over $[t_k, t_{k+1}]$.

Proof: 1. By Axiom 2: $\frac{d\mathbf{x}}{dt} = \mathbf{F}(\mathbf{x}, \mathcal{E}(\mathbf{i}_k), t) + \xi(t)$ for $t \in [t_k, t_{k+1}]$. 2. Integrating: $\mathbf{x}(t_{k+1}) - \mathbf{x}(t_k) = \int_{t_k}^{t_{k+1}} \mathbf{F}(\mathbf{x}(\tau), \mathcal{E}(\mathbf{i}_k), \tau) d\tau + \int_{t_k}^{t_{k+1}} \xi(\tau) d\tau$. 3. Defining $\xi_k = \int_{t_k}^{t_{k+1}} \xi(\tau) d\tau$ yields the result.

Definition 3 (High-Q Resonance States) A high-quality factor resonance state $\mathbf{x}^*$ satisfies:

$$\|\mathbf{F}(\mathbf{x}^*, \mathbf{0}, t)\| \leq \epsilon \quad \text{and} \quad \max |\Re(\lambda_j(\mathbf{J}_{\mathbf{F}}(\mathbf{x}^*)))| \ll 1$$

where $\mathbf{J}_{\mathbf{F}}$ is the Jacobian of $\mathbf{F}$ and λ_j are its eigenvalues.

Theorem 3 (Linear Readout Optimality) For a PRC with training pairs $\{(\mathbf{i}_k, \mathbf{y}_k)\}_{k=1}^N$, the optimal readout weights $\mathbf{W}^*$ that minimize the regularized mean squared error $\sum_{k=1}^N \|\mathcal{O}(\mathbf{x}_k) - \mathbf{y}_k\|^2 + \lambda \|\mathbf{W}\|_F^2$ are given by the solution to the ridge regression problem. For a linear readout $\mathcal{O}(\mathbf{x}) = \mathbf{W}\mathbf{x}$, the solution is:

$$\mathbf{W}^* = \mathbf{Y}\mathbf{X}^\top (\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I})^{-1} \quad (\text{A.3})$$

where $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$ is the matrix of recorded reservoir states and $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]$ is the matrix of corresponding target outputs.

Proof: 1. The regularized loss for a linear readout is:

$$\mathcal{L}(\mathbf{W}) = \|\mathbf{W}\mathbf{X} - \mathbf{Y}\|_F^2 + \lambda \|\mathbf{W}\|_F^2$$

The gradient with respect to $\mathbf{W}$ is: $\nabla_{\mathbf{W}} \mathcal{L} = 2(\mathbf{W}\mathbf{X} - \mathbf{Y})\mathbf{X}^\top + 2\lambda \mathbf{W}$.

Setting the gradient to zero, $\nabla_{\mathbf{W}} \mathcal{L} = \mathbf{0}$, yields the normal equation: $\mathbf{W}(\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I}) = \mathbf{Y}\mathbf{X}^\top$.

Solving for $\mathbf{W}$ gives the unique ridge regression solution, as the matrix $(\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I})$ is invertible for $\lambda > 0$.

Definition 4 (Computational Efficiency Metric) The energetic efficiency of a PRC is quantified by:

$$\eta_{\mathcal{R}} = \frac{\mathcal{C}(\{\mathbf{i}_k\}, \{\mathbf{y}_k\})}{E_{\text{comp}} + E_{\text{movement}}}$$

where $\mathcal{C}$ measures computational complexity, E_{comp} is energy in the reservoir, and E_{movement} is energy for data movement.

Theorem 4 (Quantum Reservoir Dimensionality) A quantum reservoir of n qubits has an effective state space dimension of:

$$\dim(\mathcal{H}_{\text{eff}}) = 2^{2n} - 1$$

accounting for the density matrix structure and trace preservation.

Proof: 1. Hilbert space $\mathcal{H}$ has dimension 2^n . 2. Density matrices ρ are positive semi-definite operators on $\mathcal{H}$ with $\text{tr}(\rho) = 1$. 3. The space of Hermitian operators on $\mathcal{H}$ has real dimension $(2^n)^2 = 4^n$. 4. The trace condition removes one degree of freedom, thus: $\dim = 4^n - 1 = 2^{2n} - 1$.

Corollary 1 (Exponential Feature Space) Quantum reservoirs provide exponential state space scaling, $\dim(\mathcal{H}_{\text{eff}}) \in O(2^{2n})$, compared to the polynomial scaling for classical reservoirs.

Proof: Direct from Theorem 4, as classical n -dimensional systems typically have state spaces scaling as $O(n^k)$ for a fixed integer k .

Definition 5 (Computational Irreducibility) A system is computationally irreducible if for most initial states $\mathbf{x}_0$, predicting its state $\mathbf{x}(t)$ requires a computation time $\tau_{\text{prediction}}$ that is of the same order as the system's physical evolution time, $\tau_{\text{evolution}}$.

Theorem 5 (Emergent Causality) In computationally irreducible systems, the causal structure perceived by a computationally bounded observer emerges from their inability to predict the system's future state faster than it unfolds. The mutual information an observer has about a future state given the present state is bounded:

$$\mathcal{I}_{\text{observer}}(\mathbf{x}(t)|\mathbf{x}(0)) \leq C_{\text{observer}} \cdot t$$

where C_{observer} is the observer's computational capacity (bits per second).

Proof: 1. By computational irreducibility, the time to predict $\mathbf{x}(t)$ is $\tau_{\text{prediction}} \geq \tau_{\text{evolution}} = t$. 2. The amount of information an observer can gain about a future state is bounded by the amount of computation they can perform, i.e., $\mathcal{I}(\mathbf{x}(t)|\mathbf{x}(0)) \leq C \cdot \tau_{\text{prediction}}$. 3. For a bounded observer, this implies: $\mathcal{I}_{\text{observer}}(\mathbf{x}(t)|\mathbf{x}(0)) \leq C_{\text{observer}} \cdot t$.

Definition 6 (Attractor States as Computation) The computational output of a physical system is defined as its convergence to an attractor state $\mathbf{x}^*$ satisfying:

$$\lim_{t \rightarrow \infty} \|\mathbf{x}(t) - \mathbf{x}^*\| \leq \delta$$

for some precision threshold $\delta > 0$.

Theorem 6 (Energy Advantage of Physical Computation) For problems with high substrate isomorphism, the ratio of energetic efficiency between a physical computer and a von Neumann machine scales as:

$$\frac{\eta_{\text{physical}}}{\eta_{\text{von Neumann}}} \in O\left(\frac{E_{\text{movement}}}{E_{\text{comp}}}\right)$$

Proof: 1. From Definition 4, efficiency is $\eta = \frac{c}{E_{\text{comp}} + E_{\text{movement}}}$. 2. For a von Neumann machine, $E_{\text{movement}} \gg E_{\text{comp}}$ (Theorem 1). Thus, $\eta_{\text{von Neumann}} \approx \frac{c}{E_{\text{movement}}}$. 3. For an ideal physical computer with co-located memory and processing, $E_{\text{movement}} \approx 0$. Thus, $\eta_{\text{physical}} \approx \frac{c}{E_{\text{comp}}}$. 4. The ratio is therefore: $\frac{\eta_{\text{physical}}}{\eta_{\text{von Neumann}}} \approx \frac{E_{\text{movement}}}{E_{\text{comp}}}$, which is a very large number for data-intensive tasks.

References

-
- Backus, J. (1978). Can programming be liberated from the von Neumann style? A functional style and its algebra of programs. *Communications of the ACM*, 21(8), 613–641. <https://doi.org/10.1145/359576.359579>
- Ghosh, S., Opala, A., Matuszewski, M., Paterek, T., & Liew, T. C. H. (2022). Quantum reservoir processing. *npj Quantum Information*, 8*(1), 83. <https://doi.org/10.1038/s41534-022-00594-4>
- Jaeger, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667), 78–80. <https://doi.org/10.1126/science.1091277>
- Landauer, R. (1961). Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3), 183–191. <https://doi.org/10.1147/rd.53.0183>
- Mead, C. (1990). Neuromorphic electronic systems. *Proceedings of the IEEE*, 78*(10), 1629–1636. <https://doi.org/10.1109/5.58356>

Nakajima, K., Fischer, I., & Tanaka, G. (2021). Reservoir computing and its applications in nonlinear signal processing. *IEEE Signal Processing Magazine, 38*(6), 86–101. <https://doi.org/10.1109/MSP.2021.3106121>

Tanaka, G., Yamane, T., Héroux, J. B., Nakane, R., Kanazawa, N., Takeda, S., ... & Nakano, D. (2019). Recent advances in physical reservoir computing: A review. *Neural Networks, 115*, 100–123. <https://doi.org/10.1016/j.neunet.2019.03.005>

Wolfram, S. (2002). *A new kind of science*. Wolfram Media.