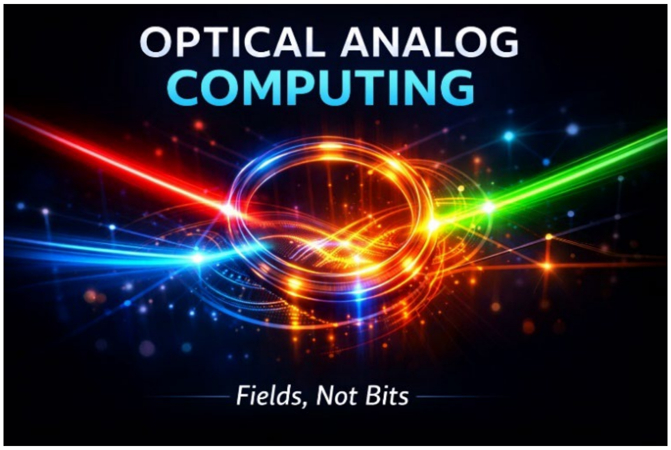

Optical Analog Computing: A Field-Based Computational Architecture

Nodes, Networks, and Field Evolution as a Computational Medium

Lev Neymotin, PhD
Samuel Neymotin, PhD

A computing paradigm in which computation emerges from the controlled evolution of physical fields.

N5Digital, Inc.

We do not compute by changing states.
We compute by shaping how a field evolves.

ABSTRACT

This work introduces a foundational architecture for an optical analog computing system in which computation emerges from the controlled evolution of electromagnetic fields. Departing from the discrete, state-based paradigm of digital computing, the proposed system operates on continuous optical fields whose amplitude, phase, wavelength, and spatial structure collectively encode information. The architecture is built around a fundamental primitive—the Node—a heterogeneously integrated resonant photonic cell in which multiple weighted optical inputs interact within a shared nonlinear cavity. Using a combination of low-loss photonic backbones, fast electro-optic control, and non-volatile phase-change materials, each Node performs programmable, nonlinear field transformations with optional memory.

Nodes are interconnected into networks that form distributed dynamical systems, where computation is realized through field propagation, interference, and convergence rather than sequential execution. The resulting computational fabric supports multiple operational regimes, including linear and nonlinear analog transformation, threshold-like behavior, and bistable dynamics. A layered sensing and readout architecture is introduced to enable observation without disrupting the computation, combining non-invasive internal monitoring, weak analog extraction, and terminal detection. A complementary programming and control framework defines how the system is configured, trained, stabilized, and operated, treating programming as the process of shaping physical parameters and guiding system evolution.

All these elements define a unified computing paradigm in which structure, fields, and dynamics are inseparable. The architecture provides a path toward scalable, parallel, and energy-efficient computation for classes of problems naturally expressed in continuous domains, including optimization, pattern recognition, and physical simulation. Rather than emulating continuous processes through discrete operations, the system computes by allowing a physical field to evolve within a programmable environment.

A central motivation for this work is the rapidly increasing energy demand of modern digital computation, particularly in GPU-centric data centers where large-scale workloads are executed through massive parallel switching of discrete states. The architecture proposed here is based on the hypothesis

that, for classes of problems naturally expressed in continuous domains, computation through controlled optical field evolution may achieve substantially lower energy consumption by replacing large sequences of transistor operations with direct physical processes. While this potential advantage is grounded in the intrinsic parallelism and low-loss propagation of optical systems, its magnitude must be established experimentally at the full system level rather than assumed a priori.

INTRODUCTION

Modern computing gained its dominance by abstracting physical reality into a disciplined symbolic world. Information became bits, operations became gates, and complexity became manageable because the machine insisted on discreteness. That choice brought extraordinary reliability, but it also imposed a cost. The physical world is not naturally discrete. Fields, waves, flows, heat, motion, and biological processes evolve continuously. Much of what digital computation does so effectively is, in a deeper sense, the emulation of continuous processes by an immense number of discrete state changes.

The system proposed here begins from the opposite side. It asks whether computation can again be rooted directly in physical continuity, not as nostalgia for classical analog machines, but as a modern, programmable, integrated architecture. The central proposition is that optical fields can form the basis of such a medium if they are allowed to evolve inside engineered structures that provide weighting, nonlinearity, memory, and controlled observation.

In free space, light is a remarkably poor computational substrate if one expects it to behave like matter. Beams may cross, interfere, and superpose, but in ordinary linear conditions they do not meaningfully modify one another. That fact is not a dead end; it is the design constraint that defines the architecture. If light is placed within carefully chosen nonlinear, resonant, and programmable photonic structures, then the optical field no longer merely passes through a passive environment. It helps shape the environment through which subsequent light propagates. Computation then appears not as symbolic switching, but as structured field evolution.

The first primitive of this system is therefore not a transistor and not a digital gate. It is a localized photonic region in which multiple optical inputs enter a common resonant state space and jointly influence its transfer behavior. In earlier formulation, this was captured as a heterogeneously integrated resonant photonic cell in which multiple weighted optical inputs couple into a common high-Q nonlinear cavity whose transfer function is dynamically tuned and persistently programmed.

That primitive becomes **Part I: the Node**. Once the Node exists, the next question is not how to build logic gates in the digital sense, but how such Nodes should be connected so that an entire optical field fabric can compute. That becomes **Part II: the Network**. Finally, because analog optical systems can be as fragile to measurement as they are rich in state, the architecture must include a disciplined philosophy of sensing and readout. Observation cannot be an afterthought. It must be built so that the act of seeing does not destroy the very computation one wants to inspect. That is described in **Part III: the Readout and Sensing Architecture**.

A practical recommendation follows from current integrated-photonics maturity. The first implementation of this system should not be optimized for visible RGB operation, even though visible multicolor operation may later become important. The initial system should be developed in the near-infrared around 1550 nm, where today's integrated photonics ecosystem is strongest. Silicon nitride has become a particularly attractive passive backbone because of its very low loss, high-Q compatibility, and broadband transparency, with recent work showing anneal-free ultra-low-loss Si_3N_4 platforms that support nonlinear resonator functions and heterogeneous integration. Thin-film lithium niobate has emerged as a powerful fast control layer because of its strong electro-optic response and demonstrated high-bandwidth modulators from visible and near-infrared through telecom-relevant regimes. Wide-bandgap phase-change materials such as Sb_2S_3 are promising for persistent multilevel weight storage because they offer low-loss non-volatile programmability in rings, Mach–Zehnders, and couplers.

A representative application of this computational paradigm is discussed in the Target Application section.

The document below treats the system not as a speculative metaphor but as a plausible integrated architecture built from functions that have already appeared, separately, in modern photonics literature. The novelty is not any single material or device. The novelty is the system logic: a computing medium in which fields, not bits, are the native objects of transformation.

The origin of this architectural perspective is outlined below.

ORIGIN OF THE ARCHITECTURE

New computational paradigms rarely emerge from entirely new physical principles. More often, they arise from a reorganization of existing knowledge—components that were previously understood in isolation are combined into a structure that reveals new behavior.

The present architecture follows this pattern.

The physical elements on which it is built—optical propagation, interference, nonlinear response, resonant structures, and programmable photonic materials—are all well established. Each has been studied extensively within its own domain: optics, photonics, materials science, and information processing. What has been largely missing is a unifying system-level perspective in which these elements are not treated as devices or effects, but as parts of a coherent computational medium.

The initial direction of this work was guided by a simple question: whether light, typically treated as a carrier of information, could instead become the **medium of computation itself**. Early considerations focused on the limits of optical confinement, beam propagation, and interaction. These questions naturally led to a fundamental constraint—light in linear media does not significantly interact with light—and therefore to the necessity of engineered nonlinearity and structured environments.

From that constraint, the architecture began to take shape. If interaction must be induced, it must occur within a controlled region. If that region is to be useful computationally, it must support weighting, transformation, and state dependence. This led to the definition of the Node as a resonant nonlinear photonic element. Once such Nodes exist, their interconnection defines a system whose behavior is no longer reducible to individual components but must be understood as a coupled field.
In this sense, the architecture is not the introduction of new physical laws, but the identification of a new way to organize known ones. It represents a shift from viewing photonic structures as signal-processing elements to viewing them as components of a computational field.

It is therefore appropriate to describe the system as both derivative and new. It is derivative in that its building blocks are drawn from established research. It is new in that their particular combination, and the computational framework built upon that combination, defines capabilities that do not emerge from the components individually.

This perspective also clarifies the intent of the work. The goal is not to replace digital computation universally, but to explore a complementary regime in which computation is performed not by discrete state transitions, but by the controlled evolution of a physical system.

RELATED WORK AND POSITIONING

Recent advances in photonic computing have demonstrated the ability of optical platforms to implement complex computational models, including photonic neural networks and neuromorphic systems with real-time learning capabilities. In these approaches, established computational frameworks—such as artificial or spiking neural networks—are mapped onto photonic hardware, leveraging optical propagation and interference to achieve high-speed, parallel execution.

While highly effective, these systems retain an algorithmic structure: computation is defined by discrete events, predefined models, and explicit update rules, with photonics serving primarily as an acceleration medium.

The architecture proposed in this work follows a different premise. Rather than implementing a predefined computational model, it defines computation in terms of the physical evolution of a coupled optical field. The system operates as a nonlinear dynamical network in which interactions, constraints, and feedback give rise to emergent behavior, and solutions correspond to stable states of the system.

This distinction places the present work in a complementary position within the landscape of photonic computing. Existing approaches extend digital and neuromorphic paradigms into optical hardware, whereas the proposed architecture explores a field-based paradigm in which computation is intrinsic to the physical system itself.

By framing computation as controlled field evolution, the architecture aligns naturally with problem classes that are continuous, constraint-driven, and dynamical, including optimization, pattern formation, and physical simulation. It therefore complements existing photonic approaches by targeting domains where solutions are more naturally expressed as equilibria rather than sequences of discrete operations.

PART I — NODE ARCHITECTURE

The Fundamental Computational Unit

1. Conceptual role of the Node

The Node is the smallest unit in which the system's computational philosophy becomes real. It is the place where multiple optical inputs cease to be merely separate beams and instead become participants in a shared local state. That local state is not purely geometric and not purely electronic. It is optical, resonant, nonlinear, and partially programmable.

A transistor enforces a controlled binary relation between input and output current or voltage. A Node does something broader. It performs a weighted transformation of multiple input fields into one or more output fields, with behavior that may be linear, nonlinear, threshold-like, or history-dependent depending on operating regime. This distinction matters. The Node is not simply an analog version of a gate. It is closer to a programmable field transformer.

That means the Node should be able to do four things at once. It must accept multiple inputs. It must allow those inputs to carry independently adjustable weights. It must expose a meaningful nonlinear response so that combined inputs can change the local transfer behavior in a nontrivial way. And it should be able, at least in some modes, to preserve part of its state so that the local response need not be wholly memoryless.

2. Material stack and division of labor

The strongest first-generation Node is not one miraculous material asked to do everything. It is a layered assembly in which each material handles the job it is best suited to perform.

The passive optical backbone should be silicon nitride. Si_3N_4 is attractive because it combines low propagation loss, compatibility with high-Q resonators, wide spectral transparency, and growing foundry maturity. Recent reports emphasize anneal-free low-temperature processes, low loss over thick geometries, and nonlinear capabilities including resonator-based parametric behavior. That makes it an appropriate skeleton for a computational medium based on propagation and resonance rather than solely on switching.

The fast control layer should be thin-film lithium niobate. TFLN has become one of the most important electro-optic materials in integrated photonics because it supports fast phase and amplitude tuning with low optical loss, and because high-bandwidth modulators and switches have now been demonstrated not only in telecom but also in visible and near-infrared regimes. In the Node, TFLN should not be asked to be the full backbone. It should instead serve as the fast analog control mechanism for tuning couplers, phase shifters, and resonant detuning.

The persistent memory and weight layer should be a low-loss phase-change material, with Sb_2S_3 an especially attractive candidate. The key benefit is not simply programmability, but non-volatile multilevel programmability with comparatively low insertion loss. Demonstrations of Sb_2S_3-clad silicon photonics have shown low loss, multilevel operation, substantial cycling endurance, and use across microrings, Mach–Zehnders, and directional couplers. In the Node, Sb_2S_3 acts as the slow but persistent structural memory: the layer that stores long-term optical weights or resonance offsets.

This division of labor leads to a useful design principle: let geometry create field enhancement, let Si_3N_4 carry the light, let TFLN tune quickly, and let Sb_2S_3 remember.

3. Physical geometry of the Node

The recommended first geometry is a bus-coupled microring resonator with tunable input coupling. The ring is not a decorative resonator added for compactness. It is the central computational chamber of the Node because it creates exactly what the architecture needs: field buildup, mode selectivity, resonance sensitivity, and the possibility of nonlinear state dependence.

A straight Si_3N_4 waveguide forms the main bus. Beside it sits a Si_3N_4 microring resonator. Light in the bus can couple into the ring when the local resonance condition permits. A second bus on the opposite side acts as a drop port, allowing the Node to support more than one observable output channel. Between the input bus and the ring, the coupler should be made tunable, ideally with a Mach–Zehnder-assisted coupling section controlled by TFLN. This tunable coupler is the Node's input valve: it determines how strongly each incoming optical field is admitted into the resonant state space.

A ring radius on the order of 30–100 µm is a reasonable conceptual first range. The important point is not the exact number but the regime. The ring should be large enough to support high-Q operation with fabricable tolerances and small enough to remain a compact computational element.
The Node should support multiple inputs. For the first-generation system, spatial multiport input is conceptually cleaner than wavelength-multiplexed input. Several waveguides approach the resonator

region from different directions, each with its own tunable coupling section and phase adjustment path. Wavelength-division approaches are certainly possible and may later yield higher density, but they also impose stronger demands on dispersion engineering, resonance alignment, and channel management. The first prototype should privilege interpretability.

4. Internal variables and computational state

Each input field can be written as $E_i = A_i e^{j\phi_i}$, where amplitude and phase are both computationally meaningful.

The simplest linear contribution can be written as:

$S = \Sigma w_i E_i$

The nonlinear response can be approximated as:

$n\_eff = n_0 + n_2 |E\_cav|^2$

Once admitted into the resonator, the effective intracavity field is not a simple sum of input powers but a coherent field result shaped by the input weights, phases, and resonant response.

The key internal variables of the Node are therefore:

- the intracavity field amplitude and phase,
- the effective refractive index experienced by the cavity mode,
- the detuning between the operating wavelength and the instantaneous cavity resonance,
- and the persistent configuration of stored weights.

These are not bookkeeping abstractions. They are the local physical state of the computational medium. The simplest linear contribution can be written as a weighted sum of input fields,

$$S = \sum_i w_i E_i$$

where the $w_i$are not abstract neural-network weights but optical quantities arising from coupler settings, phase offsets, and local detuning.

The core nonlinear relation may be modeled initially through a Kerr-type response,

$$n_{\text{eff}} = n_0 + n_2 \mid E_{\text{cav}} \mid^2$$

so that the resonance condition becomes a function of the intracavity field itself. This is the heart of the Node's computational character. The output is no longer a fixed transfer function of the inputs. It becomes a function of the input-modified local state.

5. Weighting and programmability

The Node's weights live in three different physical layers.

The first layer is geometric. Coupling gaps, ring radius, and waveguide geometry impose a fixed structural bias. These are the Node's body plan.

The second layer is dynamic and fast. TFLN-controlled phase shifters and couplers allow real-time modification of coupling coefficients and phase relationships. This layer is analogous to attention, biasing, or rapid context.

The third layer is persistent and slow. Sb_2S_3 patches placed on selected waveguide sections or ring segments can alter phase and resonance offsets in a non-volatile way. This layer gives the Node a stored personality or learned baseline.

That separation into static, fast, and persistent weights is one of the strongest architectural features of the Node. It naturally supports a two-timescale computing philosophy: slow learning or configuration, fast inference or adaptation.

6. Operating regimes

The same Node can operate in multiple distinct regimes.

In the linear analog regime, input fields are weak enough that the response remains approximately linear. The Node then behaves like a coherent weighted mixer.

In the nonlinear analog regime, higher intracavity intensities bend the transfer curve. Small changes in inputs may produce disproportionate but still continuous changes in output. This regime is essential for analog transformation, pattern shaping, and richer local dynamics.

In the threshold regime, the operating point is placed near a steep resonance slope so that a modest cumulative input shift can move the Node abruptly between low and high transmission states. In this mode, the Node can mimic digital-like logic, but this is only one operating mode, not the Node's core identity.

In the bistable regime, the interplay of nonlinearity and detuning allows two stable states to coexist for the same external conditions. This is the Node's entry point into memory. The Node then ceases to be merely a mapper of present inputs and becomes sensitive to history.

7. Boundaries of the first Node design

The first Node should be passive, programmable, nonlinear, and optionally bistable, but it should not include on-node gain. Gain complicates thermal behavior, noise, and architectural clarity. Platforms such as InP are excellent for active photonics and will likely play a major role in the larger system, but gain should initially be treated as an infrastructure layer placed between subnetworks rather than inside the primitive itself. This separation protects the conceptual purity of the Node and simplifies early experimental validation.

8. Formal identity

The Node can therefore be defined as follows:

A Node is a heterogeneously integrated resonant photonic cell in which multiple weighted optical inputs couple into a common high-Q nonlinear cavity, whose transfer function is dynamically tuned by fast electro-optic control and persistently programmed by non-volatile phase-change elements.

That definition is technical, but its meaning is simple. The Node is the place where the field becomes locally self-referential. Inputs do not merely traverse it. They help determine what it is.

A schematic representation of the Node is shown in Figure 1.

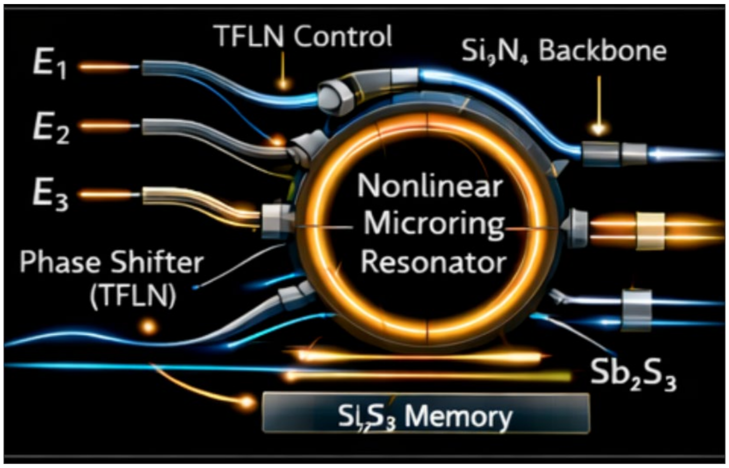

Figure 1 — Node Architecture: Programmable Nonlinear Resonant Photonic Cell
A Node is implemented as a bus-coupled microring resonator with multiple independently controlled input channels. Each input field is weighted through tunable coupling and phase control before entering a shared nonlinear cavity. The intracavity field induces a state-dependent response through nonlinear interaction, producing output fields at the through and drop ports. The Node integrates low-loss photonic propagation, fast electro-optic tuning, and persistent phase-change programming, forming the fundamental computational unit of the system.

PART II — NETWORK ARCHITECTURE
The Optical Analog Computational Fabric

1. From device to medium
A single Node is an interesting photonic element. A network of Nodes is a computational medium.

Digital systems derive power from composability: gates can be connected into larger circuits without changing the meaning of the individual gate. In the optical analog architecture proposed here, composability takes a different form. Nodes do not merely transmit symbols downstream. They exchange fields, and the network's computational character emerges from the coupled evolution of those fields across space and time.

The correct mental shift is therefore not from gate arrays to larger gate arrays, but from isolated field transformers to a distributed dynamical system.

2. Coupling as the grammar of computation
The essential network equation is

$$E_i^{\text{next}} = f_i\left(\sum_j C_{ij}\, E_j,\ W_i\right)$$

where C_{ij} represents the optical coupling between Nodes and W_i represents the internal programmable parameters of Node i.

This equation is more revealing than it looks. It says that the network is not defined only by what each Node is, but by how each Node is embedded into a field topology. The couplings are not wires in the digital sense. They are weighted field relationships.

There are several important coupling classes.

Directed coupling corresponds to guided propagation from one Node to another along a specific path. It is the closest analog to a wire and is useful where computational structure must be explicit.

Broadcast coupling allows one Node to influence many others. This is valuable when a common control field, reference pattern, or global context must be distributed.
Interference coupling allows multiple outputs to combine before entering a destination Node. This means part of the computation may occur in the path itself, before the receiving Node's own nonlinearity engages.

Feedback coupling closes loops. This is essential for memory, recurrent dynamics, oscillation, stabilization, and attractor formation. In an analog optical computer, feedback is not an accessory. It is one of the main sources of computational richness.

3. Topological families
A feedforward fabric is the simplest topology. Inputs enter one side of the network, propagate through successive layers of Nodes, and emerge as transformed outputs. This regime is useful for stable analog mapping, inference, static transforms, and controlled function approximation.

A recurrent fabric contains loops. The network must then be understood in time, not just in space. Such networks can settle to fixed points, oscillate, or traverse rich transient trajectories. This regime is central for associative memory, optimization, pattern completion, and temporal processing.

A lattice or field fabric arranges Nodes in a spatial grid with predominantly local couplings. In this regime the network begins to resemble a discretized physical medium. It can naturally emulate diffusion, wave propagation, relaxation, or other partial-differential-equation-like dynamics. This makes it particularly attractive for optimization and physical simulation problems.

A hybrid topology will likely be the most powerful practical architecture. Feedforward sections can prepare or preprocess signals, recurrent cores can perform dynamic convergence, and lattice-like subnetworks can solve spatial analog problems.

4. Representation of information
One of the great opportunities of the system is that a signal is not just one scalar value. In a digital computer, a voltage level or logic state is usually stripped of all richness except its binary meaning. In this architecture, information may live simultaneously in amplitude, phase, wavelength, spatial mode, and temporal modulation.

That does not imply that every network should always use all available degrees of freedom. In fact, disciplined sub-architectures will often restrict the signal space in order to maintain robustness. But the native representational capacity of the medium is high-dimensional. This is one of the reasons the system should not be thought of as "values between 0 and 1." It is a field architecture, not a scalar analog circuit.

5. Computation as convergence or evolution

A feedforward network computes by producing an output field from an input field. A recurrent or lattice network may compute by settling. In such cases, the answer is not an immediate output but a converged or characteristic state of the fabric.

This distinction is profound. Digital computers generally compute by executing an ordered sequence of operations. The optical analog fabric computes by letting a coupled physical system evolve under structured constraints. In this sense, the architecture is closer to a designed physical process than to a conventional instruction stream.

That shift may prove especially important for optimization, inverse problems, analog inference, and physical simulation, where the desired answer is often the stable state or low-energy configuration of a dynamical system.

The process of computation as field evolution and convergence within the optical network is illustrated schematically in Figure 2.

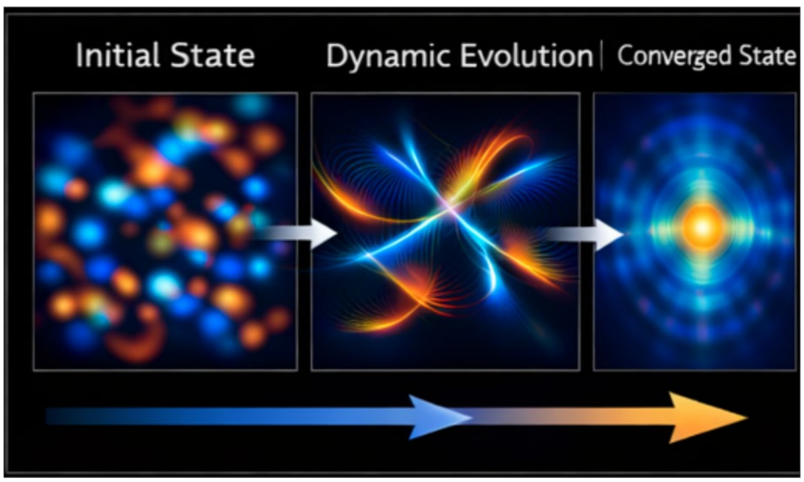

Figure 2 — Field Evolution and Convergence in an Optical Analog Network

Computation in the proposed architecture occurs through the evolution of a coupled optical field. An initial input configuration (left) propagates through the network, where nonlinear interactions and feedback progressively shape the field (center). The system converges toward a stable configuration (right), representing the solution to the imposed constraints. Unlike digital computation, which proceeds through discrete steps, the solution emerges as the equilibrium state of the system.

6. Gain, loss, and energy management

Because fields propagate continuously, loss management is unavoidable. Waveguide loss, coupler inefficiency, scattering, and nonlinear absorption all erode signal strength. A large network cannot function on passive propagation alone unless it is extremely shallow or deliberately loss-tolerant.

For this reason, gain should enter at the system level even if it is excluded from the primitive Node. InP remains one of the most mature integrated active photonics platforms and is a natural candidate for amplifier or source subsystems. The core principle is architectural separation: allow the Nodes to remain conceptually clean while letting external gain stages restore signal budgets at designated points.

Energy management in this system is therefore not just a power-supply problem. It is part of the computational architecture. Gain placement, saturation behavior, and normalization are all computationally relevant because they determine whether the fabric remains in a usable operating regime.

7. Stability and control

A field-based analog medium is powerful precisely because it is continuous. That same continuity makes it vulnerable to drift, crosstalk, thermal detuning, and runaway dynamics. Stability mechanisms therefore have to be built into the network from the start.

At the local level, resonant filtering and nonlinear saturation help keep Node responses bounded. At the mesoscopic level, feedback control and tuning loops can maintain resonance alignment and coupling balance. At the global level, normalization layers or control buses may be required to keep the entire network on a computationally useful manifold rather than drifting into chaos or silence.

An architecture that ignores stabilization will appear elegant on paper and fail in practice.

8. Programming and reconfiguration

The network should be programmable at more than one level.
Structural programming is done through layout: which Nodes exist, how they are connected, what coupler geometries and path lengths are chosen.

Fast dynamic programming is done through the TFLN layer: phase, coupling, resonance bias, and perhaps temporal modulation can be changed in real time.
Persistent programming is done through the phase-change layer: long-term weights, calibration offsets, or learned configurations can be written and retained without continuous power.

This multi-layer programmability is a major strength of the system. It gives the network both a body plan and a nervous system.

9. Identity of the Network

The Network is therefore best understood not as a circuit in the conventional sense but as a self-evolving optical field system whose computational behavior emerges from the interplay of Node physics, coupling topology, and controlled energy flow.

PART III — READOUT AND SENSING ARCHITECTURE

Observation Without Destroying the Computation

1. Why measurement must be architected

A strong analog optical fabric can be damaged more easily by bad measurement than by imperfect design. A sensor that simply extracts light from every interesting point may render the network legible while simultaneously degrading the very amplitudes and phases that encode the computation. Reflection from measurement structures can corrupt resonant behavior. Nearby electronics can thermally detune sensitive Nodes. Noise introduced by the measurement system can propagate backward into the control layer.

This is not a hypothetical concern. In dense photonic integrated circuits, conventional tap-and-detect monitoring scales poorly because every tap steals optical power. That is one of the reasons non-invasive monitoring methods such as CLIPP were developed. The CLIPP family monitors guided optical power indirectly through conductance variation rather than direct optical tapping, specifically to reduce disturbance in large-scale integrated photonics.

The architecture should therefore distinguish between three very different acts: monitoring internal state, observing analog results, and harvesting final outputs. These should not be conflated.

2. Layer 1: transparent internal monitoring
The innermost sensing layer should consist of transparent observers whose job is not to read the answer but to help the network remain itself. These monitors track local optical power, resonance occupancy, drift, and branch balance while adding negligible optical loss.

CLIPP-type monitors are a strong model for this layer because they can provide non-invasive on-chip optical monitoring and are explicitly intended for dense integrated photonics. In the present architecture, such monitors belong near resonant Nodes, reconfigurable couplers, selected feedback loops, and calibration-critical branches.

These sensors form the network's proprioception. They tell the system how it is behaving internally, but they do not constitute final readout.

3. Layer 2: weak analog observation
The second sensing layer should consist of deliberately chosen observation points from which small fractions of optical signal are extracted into dedicated observation buses. These points are not placed everywhere. They are sparse, intentional, and architecturally separated from the deepest recurrent interior.

Once a weak sample is available, the preferred readout method depends on the information one seeks. If the relevant variable is intensity only, direct analog photodetection may be sufficient. If phase, coherent superposition, or interference state matters, the readout must be coherent.
Integrated coherent receivers are therefore highly relevant to the architecture. Recent integrated photonic-electronic coherent receiver work illustrates that compact, sensitive, balanced coherent readout is feasible and can approach very strong noise performance. In the context of this system, coherent observation is not a luxury. It is the correct measurement language whenever the computation itself is field-based rather than intensity-only.

This layer is where the network becomes interpretable without being fully terminated.

4. Layer 3: terminal output readout
At the boundary of the computation, the system is allowed to be destructive. Terminal readout converts optical output states into electronic signals for logging, control, storage, or human interpretation.

The specific detector technology depends on the wavelength band. For visible and near-visible platforms, integrated waveguide-coupled photodetectors on SiN-related stacks are developing rapidly, though the ecosystem remains less mature than around telecom wavelengths. Reviews of visible and near-visible integrated photonics explicitly note progress in modulators, couplers, and waveguide-coupled detectors on foundry-compatible platforms. For the near-infrared starting point recommended here, the detector environment is broader and more mature.

This terminal layer sits outside the core analog recursion. Its job is to translate, not to preserve.

5. Back-action and its suppression
The readout layer must be engineered against several forms of back-action: optical depletion, back-reflection, thermal disturbance, carrier perturbation, and electrical-noise injection.
Transparent internal monitors minimize depletion. Observation buses isolate stronger measurements from the computational interior. Coherent detection reduces the need for large extracted optical power because a local oscillator enhances sensitivity. Balanced detection helps suppress common-mode noise in the electrical domain. And physical separation between sensitive resonant regions and readout electronics limits thermal and electrical interference.

This is why the architecture should be described not merely as “gentle sensing,” but as a hierarchy of observation. The act of seeing is itself structured.

6. Temporal strategy of observation
Not every question about the network should be asked at the same speed. Fast, shallow observations should track local health: Is the ring still near its intended operating point? Has a branch drifted out of balance? Is a feedback loop oscillating abnormally?

Slower, deeper observations should interpret the computation: Which attractor did the recurrent subnetwork settle into? What analog value emerged at a selected subnet boundary? What phase relation distinguishes one solution class from another?

This dual-timescale approach matches the architecture naturally. Transparent monitors support ongoing stabilization. Weak analog observers support interpretation. Terminal detectors harvest final outcomes.

7. Identity of the Readout Architecture
The readout architecture is therefore not a collection of detectors attached to a photonic network. It is a layered observation system designed so that monitoring, interpretation, and final harvesting are physically separated but optically coordinated.

System Synthesis
The system described in Parts I–III can be summarized by a unified conceptual equation:

$$E_i^{\text{next}} = f_i\left(\sum\nolimits_j C_{ij}\, E_j,\ \ |\, E_{\text{cav}}\,|^2,\ W_i\right)$$

where the inputs and outputs are fields, the couplings are optical relationships, the local state is resonant and nonlinear, and the weights are distributed across geometric, dynamic, and persistent layers.

The Node gives the system a local transformation law.
The Network gives it distributed evolution.
The Readout Architecture gives it disciplined visibility.

Combined, they define a computing medium in which physical fields, rather than symbolic bits, are the native objects of manipulation.

PART IV — PROGRAMMING, TRAINING, AND CONTROL FRAMEWORK
From Physical Medium to Operational Computing System

1. From Architecture to Operation
With the Node, the Network, and the Readout architecture defined, the system now exists as a coherent physical construct. It can transform fields, propagate interactions, and be observed without collapsing its internal state. Yet at this stage, it is still only a medium—rich, structured, and potentially powerful, but not yet a computer in the practical sense.

The transition from medium to machine occurs here.
In digital systems, programming is conceptually separate from physics. Instructions are written in symbolic form, compiled, and executed by hardware whose internal structure remains largely unchanged during operation. The program is external; the machine is fixed.

In this system, that separation does not hold.

Programming is the act of configuring a physical field system so that its natural evolution produces a desired outcome.

There is no strict boundary between program and execution. The configuration of weights, the geometry of connections, and the injected input fields together define both the problem and the process by which it is solved. Computation is not a sequence of steps imposed on hardware. It is the behavior of the hardware itself under controlled conditions.

This demands a different conceptual framework—one in which programming, training, and control are not discrete stages but interdependent aspects of guiding a continuous system.

2. Programming as a Multi-Layer Physical Process
Programming in this architecture unfolds across multiple layers, each corresponding to a different level of permanence and timescale. These layers are not abstractions; they are physically embodied in the materials and structures defined in Part I.

At the deepest level lies the geometry of the system. The arrangement of Nodes, the topology of their connections, and the physical layout of waveguides and couplers together define what computations are even possible. This structural layer is fixed at fabrication and acts as the system's "computational landscape." A feedforward topology predisposes the system toward static mappings, while recurrent or lattice-like structures enable dynamic evolution, convergence, and spatial computation. In this sense, geometry is not merely implementation—it is the first and most fundamental program.

Above this structural layer lies persistent configuration. This is realized through non-volatile phase-change materials such as Sb_2S_3 (Antimony Sulfide), which allow local optical properties to be modified and retained without continuous power. These modifications alter resonance conditions, coupling strengths, and phase relationships, effectively encoding long-term weights into the system. Unlike digital memory, which stores symbolic values, this layer stores **physical bias conditions** that directly shape how the system behaves. It is here that training outcomes are embedded and preserved.

The third layer is dynamic control, implemented through fast electro-optic tuning using materials such as Thin-Film Lithium Niobate (TFLN). This layer operates on timescales far shorter than those of persistent programming and allows the system to respond in real time. Phase shifts can be applied, coupling ratios adjusted, and resonances fine-tuned dynamically. If the persistent layer defines the system's long-term character, the dynamic layer provides its moment-to-moment adaptability.

These three layers—structural, persistent, and dynamic—form a hierarchy of programming. Together, they define a system that is not merely configurable but **continuously reconfigurable**, with different aspects of its behavior governed at different temporal scales.

3. Inputs as Active Elements of the Program
In conventional computation, inputs are treated as passive data to be processed. In this architecture, inputs are active participants in the computation itself.

Each input is an optical field characterized not only by amplitude, but also by phase, wavelength, and temporal structure. These properties are not incidental—they are integral to the computation. A change in phase can alter interference patterns. A change in amplitude can shift a Node from linear to nonlinear operation. A change in wavelength can engage or bypass specific resonances.

The input field is part of the program.

This creates a dual-layer programming model. The internal configuration of the system defines its general behavior, while the input field defines the specific computation to be performed at any given moment. In some cases, changing the input alone is sufficient to redirect the computation entirely, without modifying any internal parameters. This ability to encode instructions directly into the physical signal is one of the system's defining strengths.

4. Training as Physical Adaptation
Training in this system is not the adjustment of abstract parameters in software. It is the process of modifying the physical configuration of the system so that its response to inputs aligns with desired outcomes.

Before any training can occur, the system must be calibrated. Calibration establishes a known operating point by aligning resonances, normalizing coupling strengths, and compensating for fabrication variability and environmental drift. This step is essential because the system's behavior is highly sensitive to its physical state. Without calibration, training becomes unpredictable and results are not reproducible.

Once calibrated, the system can be trained through iterative adjustment. A typical training cycle begins with the application of a known input field. The resulting output is measured using the readout architecture defined in Part III. The difference between the observed output and the desired output is then used to guide adjustments to the system's parameters. These adjustments may involve modifying persistent weights via phase-change programming or fine-tuning dynamic parameters through electro-optic control.

This process resembles gradient-based optimization, but the gradients are not computed symbolically. Instead, they are inferred through physical perturbation and observation. Small changes are applied, the resulting effect is measured, and the system is nudged in the direction that reduces error. In this sense, the system participates directly in its own training.
In certain network topologies—particularly those involving feedback or spatial coupling—training can take on a more intrinsic form. Rather than explicitly minimizing an error function, the system can be configured so that desired solutions correspond to stable states of the network. Training then becomes

the process of shaping the system's energy landscape so that correct solutions emerge naturally as attractors. This is especially powerful for problems involving optimization, pattern recognition, or associative memory, where the goal is not to compute a function but to settle into a meaningful configuration.

In practice, a hybrid approach is likely to be most effective. Digital control systems can guide the training process, while the optical system provides fast, parallel evaluation of candidate configurations. This "hardware-in-the-loop" approach leverages the strengths of both domains: the precision and flexibility of digital computation and the speed and parallelism of analog field evolution.

5. Control as Continuous Stabilization

Because the system operates on continuous physical variables, it is inherently sensitive to drift, noise, and environmental variation. Control is therefore not an auxiliary function—it is essential to maintaining computational integrity.

The control system operates across multiple timescales. At the fastest level, it stabilizes phase and resonance conditions using feedback from internal sensors. These adjustments occur on nanosecond to microsecond timescales and ensure that the system remains within its intended operating regime. At intermediate timescales, the control system compensates for thermal effects and maintains coupling balance across the network. At the slowest level, it updates persistent weights and oversees the training process.

These control loops rely on the sensing architecture defined in Part III. Transparent internal monitors provide continuous information about local optical power and resonance conditions. Weak analog observers supply more detailed information at selected points. Together, they allow the control system to maintain stability without significantly disturbing the computation.

The goal of control is not merely to keep the system running, but to ensure that it behaves reproducibly and converges reliably. In a system where computation is defined by physical evolution, control defines the conditions under which that evolution is meaningful.

6. Execution as Field Evolution

Execution in this architecture does not consist of a sequence of discrete operations. It is the process by which the configured system evolves under the influence of an input field.

In its simplest form, execution proceeds as follows. An input field is applied to the network. The system responds according to its current configuration, and the optical field evolves through the network of Nodes. After a transient period, the system may settle into a steady state. The resulting field is then measured at the output boundary.

In more dynamic scenarios, the input may vary continuously, and the system's response evolves in time. The output is then not a single value, but a trajectory—a time-dependent field that reflects the ongoing interaction between input and system.

In iterative scenarios, the system may be driven through a sequence of inputs or control adjustments, with each step building on the previous state. This is particularly useful for optimization and search problems, where the solution emerges gradually through successive refinements.

In all cases, execution is not imposed from outside. It is the natural behavior of the system under controlled conditions.

7. Noise, Error, and Physical Precision

No physical system is free of noise, and this architecture is no exception. Thermal fluctuations, optical shot noise, and fabrication imperfections all introduce variability into the system's behavior.

The appropriate response is not to attempt to eliminate noise entirely, but to design the system so that it is robust to it. This can be achieved through redundancy, averaging, and the use of attractor-based computation, where the system's stable states are resilient to small perturbations.

Precision in this system is continuous but bounded. It is not limited to discrete levels, but it is constrained by physical factors such as signal-to-noise ratio and device stability. For many applications—particularly those involving approximation, optimization, or pattern recognition—this level of precision is not only sufficient but advantageous, as it allows the system to operate efficiently without the overhead of maintaining strict digital exactness.

8. Verification and Validation

Verification in this architecture must occur at multiple levels. At the local level, individual Nodes and couplings must be characterized to ensure that they behave as expected. At the global level, the behavior of the network as a whole must be validated against known inputs and outputs.

The sensing architecture plays a central role in this process. Internal monitors verify that the system remains within its intended operating regime. Analog observers provide insight into intermediate states. Terminal readout confirms final results.

Because the system is adaptive and reconfigurable, verification is not a one-time process. It must be ongoing, integrated into the control framework, and responsive to changes in configuration and environment.

9. Reconfiguration and Adaptation

One of the most powerful features of the system is its ability to be reconfigured. Through a combination of persistent programming and dynamic control, the same physical network can be adapted to perform different tasks.

Reconfiguration may involve rewriting phase-change states to alter long-term weights, adjusting electro-optic controls to modify coupling and phase relationships, or simply changing the input field to redirect the computation. In more advanced modes, the system may adapt continuously, updating its configuration in response to environmental feedback or performance metrics.

This adaptability allows the system to function not as a fixed-purpose device, but as a **general computational medium**.

10. Operational Modes

The system naturally supports multiple modes of operation.

In exploration mode, the system scans its parameter space, identifying stable regions and useful configurations. In training mode, it adjusts its parameters to align its behavior with desired outcomes. In inference mode, it operates with a fixed configuration, performing computations rapidly and efficiently. In adaptive mode, it continuously updates its parameters in response to changing conditions.

These modes are not mutually exclusive. The system may transition between them as needed, or operate in hybrid modes where training and inference occur simultaneously.

11. Programming Identity
The programming framework can be summarized in a single statement:

Programming is the process of configuring and guiding a physical field system so that its natural evolution produces desired computational behavior.

This is not programming in the conventional sense. It is closer to shaping a landscape than writing a script.

The implications of this computational model become most evident when applied to real-world systems in which solutions correspond to dynamically maintained equilibrium states.

TARGET APPLICATION — REAL-TIME ANALOG OPTIMIZATION VIA PHYSICAL CONVERGENCE
The computational model introduced in this work is fundamentally aligned with problems in which solutions emerge not through discrete calculation, but through the convergence of a system toward a stable configuration. In such problems, the desired outcome is not a sequence of intermediate steps, but a final state that satisfies a set of constraints.

The proposed optical analog computing system is naturally suited to this class of problems because it is itself a dynamical system. Rather than iteratively computing candidate solutions, the system evolves continuously under the influence of input conditions and internal interactions. The solution corresponds to the equilibrium state of this evolution.

This mode of operation is particularly well matched to **real-time analog optimization problems**, where a system must continuously adapt to changing conditions while maintaining global consistency across many interacting variables. In such scenarios, the optical field does not represent data in a traditional sense. It represents the **state of the system**, and computation proceeds through its evolution toward a stable configuration.

Unlike digital approaches, which must simulate such processes through discrete steps, the optical system performs the optimization directly through its physical behavior. The network of Nodes collectively enforces constraints, while nonlinear interactions and feedback drive the system toward convergence.

This approach enables a form of computation in which:

- constraints are embedded physically,
- interactions occur in parallel,
- and solutions emerge organically as stable states.

Example — Real-Time Power Grid Stabilization
A concrete example of such a problem is the real-time stabilization of an electrical power grid.

Modern power grids must continuously balance supply and demand across large, interconnected networks. This involves:

- matching generation to consumption
- maintaining frequency stability
- routing power through constrained transmission paths
- responding to disturbances such as load fluctuations or generator failures

These constraints are strongly coupled and continuously changing. The system must find a configuration that satisfies all constraints simultaneously and adapt to new conditions in real time. Such systems are traditionally solved using iterative numerical methods that approximate equilibrium through repeated computation.

In conventional digital systems, this is addressed through iterative numerical optimization, often requiring repeated recalculation and significant computational resources.

In contrast, the proposed optical analog computing system can represent the grid as a network of interacting Nodes, where:

- each Node corresponds to a local element of the grid (e.g., generator, load, or junction),
- coupling between Nodes represents transmission constraints,
- input fields encode current system conditions, and
- nonlinear interactions enforce physical limits and balance conditions.

As the system evolves, the optical field naturally settles into a configuration that satisfies the global constraints. This configuration corresponds to a stable operating point of the grid.
If conditions change—for example, due to a sudden load increase—the input field changes, and the system evolves again, settling into a new equilibrium without requiring explicit recalculation.

In this sense, the system behaves analogously to a physical network seeking equilibrium:
it continuously adjusts itself until balance is restored.

The mapping of a real-time power grid stabilization problem onto the optical analog computing system is illustrated schematically in Figure 3.

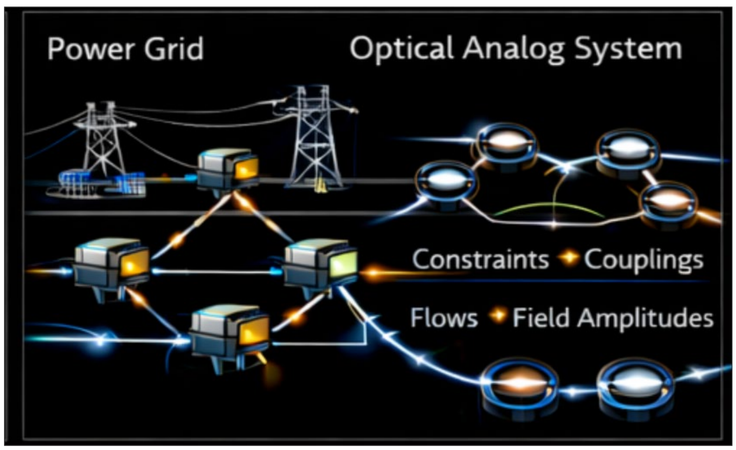

Figure 3 — Mapping of Power Grid Optimization to Optical Analog Computing System
A real-time power grid can be represented as a network of interacting constraints (left), including generation, load distribution, and transmission limits. In the proposed architecture (right), these elements are mapped onto a network of photonic Nodes, where coupling represents transmission constraints and optical fields encode system state. The system evolves continuously, and the stable optical field configuration corresponds to a balanced operating point of the grid. This illustrates how real-world optimization problems can be solved through physical convergence rather than iterative computation.

This example illustrates the essential advantage of the architecture: it does not compute solutions step by step, but **allows the solution to emerge through physical convergence**.

CONCLUSION
This document has presented a complete architectural framework for an optical analog computing system grounded in physical field dynamics. Beginning with the definition of the Node as a programmable nonlinear resonant photonic cell, and extending through network organization, sensing, and operational control, the system has been developed as a coherent computational medium rather

than a collection of isolated components. Each part contributes a necessary dimension: local transformation, global interaction, disciplined observation, and controlled evolution.

The central departure from digital computing lies in the unification of program, execution, and physical state. In this system, computation is not imposed externally through symbolic instruction, but arises from the behavior of a configured physical structure under the influence of input fields. Programming becomes the act of shaping that structure and its parameters; execution becomes the evolution of the field; and the result is the stabilized or observed state of the system. This integration allows the architecture to exploit the inherent parallelism and continuity of optical fields while retaining programmability and control.

The proposed design leverages existing advances in integrated photonics—low-loss waveguides, high-Q resonators, electro-optic tuning, and phase-change materials—combined in a novel system-level configuration. While significant engineering challenges remain, particularly in large-scale integration, stability, and precise control, the underlying components and principles are grounded in demonstrated technologies. The architecture therefore represents not a speculative departure, but a synthesis of known elements into a new computational form.

The potential impact of such a system lies in its alignment with problems that are inherently continuous, high-dimensional, and parallel. By computing through physical evolution rather than symbolic iteration, the system may offer new efficiency regimes for optimization, inference, and simulation tasks that are costly or unnatural in digital frameworks.

One of the principal motivations underlying this architecture is the emerging energy constraint of digital computing at scale. As modern workloads increasingly rely on large clusters of GPUs operating in highly parallel regimes, total system power consumption is approaching levels that challenge both economic and physical sustainability. The optical analog computing system proposed here offers a fundamentally different path: rather than performing computation through vast numbers of discrete switching events, it seeks to realize computation as the controlled evolution of a physical field. This shift introduces the possibility of significantly improved energy efficiency for appropriate problem classes. However, the extent of such gains must be validated at the system level, taking into account not only the photonic core but also control, stabilization, and readout layers. The question is therefore not whether optical analog systems can replace digital computation universally, but where their physical operating principles provide decisive advantages.

A detailed assessment of constraints and development challenges is provided in Appendix A.

A representative application of this computational paradigm is discussed in the Target Application section.

Future work should focus on three directions. First, the development of application-specific network topologies that demonstrate clear advantages over digital approaches. Second, the refinement of training and control methodologies, including hybrid digital-analog optimization loops. Third, the experimental realization of small-scale prototypes that validate the Node and its integration into functional networks.

Ultimately, the architecture invites a reconsideration of computation itself—not as a sequence of discrete operations, but as the controlled evolution of a physical system.

The question is no longer how to make digital systems larger, but whether computation itself can be redefined in a more physically efficient form.

The system does not compute the state of the system—it becomes the state of the system.

APPENDIX A — CRITICAL ASSESSMENT, CONSTRAINTS, AND DEVELOPMENT PATH

Technical Limits, System Challenges, and Practical Viability

Scope and Purpose of the Assessment

The architecture presented in this work defines a coherent and physically grounded approach to optical analog computing. It is constructed from well-established physical principles and supported by advances in integrated photonics, nonlinear optics, and programmable materials. At the same time, its viability as a practical computing system must be assessed critically, both from the standpoint of fundamental physics and from the perspective of engineering implementation at scale. The purpose of this appendix is to examine the principal constraints, limitations, and risks associated with the proposed system, and to clarify a realistic path forward.

Interaction Limits and the Role of Nonlinearity

A central premise of the architecture is that computation emerges from the interaction of optical fields within engineered structures. In linear media, however, light does not significantly interact with light. This is not merely a technical inconvenience; it is a defining constraint. Meaningful computation therefore depends on the introduction of controlled nonlinearity, achieved in the proposed design through resonant enhancement and material response within the Node. While this approach is physically valid and widely studied, it places the system within a narrow operating regime. High-quality-factor resonators amplify field interactions, but they also introduce sensitivity to fabrication variation, temperature fluctuations, wavelength detuning, and cross-coupling between neighboring elements. As a result, the computational mechanism is inherently tied to conditions that must be actively maintained. The architecture is therefore viable, but not intrinsically robust without additional stabilization.

Analog Precision and Accumulated Uncertainty

This sensitivity leads directly to a second class of constraints associated with analog precision. The system operates on continuous optical fields, which provide expressive computational power but also inherit the limitations of analog systems. Thermal noise, shot noise, phase instability, and device mismatch all contribute to variability in system behavior. Unlike digital systems, where discrete states provide natural error tolerance, analog systems must manage such variability explicitly. Precision is therefore continuous but bounded, and the accumulation of small errors across networks becomes a real concern. This does not invalidate the architecture, but it does define its natural domain of applicability. The system is well aligned with problems that tolerate approximation—such as optimization, pattern recognition, and physical simulation—but less suited to tasks requiring strict numerical exactness and reproducibility.

Control Complexity and Stabilization Requirements

Closely related to this issue is the role of control and stabilization. The proposed Node relies on resonant and nonlinear effects that are highly sensitive to environmental conditions. Maintaining stable operation therefore requires continuous monitoring and adjustment of phase, resonance, and coupling parameters. This introduces a control layer that is essential to system functionality. However, it also introduces a potential imbalance: the infrastructure required to stabilize the system may approach the complexity of the computational core itself. The architecture explicitly includes such control mechanisms, but their cost—in terms of energy, hardware, and system design—must be carefully managed. Without effective control strategies, the system may drift out of its intended operating regime, undermining both performance and reproducibility.

Persistent Memory and Programmability Constraints

Another important consideration is the implementation of memory and programmability. The architecture relies on phase-change materials such as Sb_2S_3 to provide non-volatile, multilevel weight storage. This is a promising approach that aligns naturally with the physical nature of the system. However, phase-change photonic devices remain an active area of research and carry practical limitations, including finite endurance, non-negligible write energy, device variability, and potential drift over time. In addition, the architecture does not yet offer a direct equivalent to the large-scale, high-bandwidth memory systems that underpin modern GPU-based computation. This limits the system's suitability for problems requiring extensive data movement or large external memory, and suggests that its initial applications will be those in which computation can be expressed primarily through locally embedded weights.

Interface Overhead and System Boundary Costs

A further constraint arises at the interface between optical and electronic domains. Any practical system must perform electrical-to-optical conversion, optical-to-electrical conversion, and signal conditioning through digital-to-analog and analog-to-digital processes. These interfaces introduce both energy overhead and system complexity. A key risk is that the energy required for conversion, control, and readout may dominate the total system energy, reducing the advantage of the optical core. This is not a limitation of the optical computation itself, but of the surrounding infrastructure required to realize it as a practical system.

Fabrication, Integration, and Packaging Challenges

Fabrication and integration present additional challenges. The architecture requires the combination of multiple technologies, including low-loss waveguides, nonlinear resonators, electro-optic control layers, phase-change materials, and detectors, all within a single platform. While each of these elements has been demonstrated individually, their integration at scale remains complex. Photonic packaging, in particular, is a well-known bottleneck, involving precise optical alignment, thermal management, and often significant manual intervention. These factors impact not only cost but also reproducibility and scalability. The system is therefore feasible as a research platform, but large-scale manufacturing remains a significant challenge.

Energy Efficiency: Potential and Uncertainty

One of the principal motivations for this work is the increasing energy demand of modern digital computation. The proposed architecture offers a credible path toward improved energy efficiency by replacing large numbers of discrete switching events with continuous physical processes. However, the magnitude of this advantage cannot be assumed in advance. A complete energy assessment must include not only the optical core but also sources, control electronics, stabilization mechanisms, memory programming, and readout systems. It is entirely possible for the optical computation to be highly efficient while the overall system achieves only modest gains due to surrounding overhead. The energy advantage, if it exists, will therefore be highly dependent on the specific application and system design.

Scope of Applicability

Taken together, these considerations define both the limitations and the opportunity of the architecture. The system should not be viewed as a universal replacement for digital computing. Its strengths lie in problems that are continuous, high-dimensional, and naturally expressed as field evolution. In such domains, it may offer capabilities that are difficult or inefficient to achieve with digital methods. Conversely, for tasks requiring exact symbolic manipulation or high-precision arithmetic, digital systems will remain dominant.

Development Path Forward

A realistic path forward follows from this assessment. The first stage is the validation of the Node as a stable, programmable nonlinear element. This must be followed by the demonstration of small networks performing well-defined analog tasks, where the behavior of the system can be measured and understood. Only then can meaningful benchmarking be performed against conventional approaches, focusing on latency, energy, and accuracy for specific problem classes. From there, development can proceed toward application-specific systems in which the advantages of the architecture are clearly demonstrated.

Summary and Practical Outlook

In summary, the optical analog computing architecture proposed in this work is physically plausible and conceptually coherent, but its practical realization requires overcoming significant challenges across physics, engineering, and system integration. It should be understood not as an immediate replacement for existing digital systems, but as a promising framework for specialized computation in domains where its underlying principles provide a natural advantage.