Intelligence, Consciousness and Designing for Computational Efficiency

Peter David Fagan, School of Informatics, University of Edinburgh

April 2, 2026

Abstract

We present a physical framework for computational efficiency in AI by mapping classical complexity to informational costs measured in *nats*. Algorithmic time is treated as irreversible information processing (I_{irr}), while space is treated as preserved information processing (I_{rev}). Using these quantities, we define operational intelligence ($\chi = W_{\text{achieved}}/I_{irr}$), structural consciousness ($\kappa = W_{\text{achieved}}/I_{rev}$), and retention ($\rho = W_{\text{achieved}}/W_{\text{target}}$) to separate task demand from architecture-level performance. We ground these metrics in lightweight Conservation-Congruent Encoding (CCE) conditions, which specify when coarse-grained informational states admit metastable physical realizations and when irreversible state collapse must export entropy through conserved channels. We then apply the framework to sorting algorithms and modern sequence models, showing that dominant scaling bottlenecks are physical routing burdens rather than software abstraction alone. Under this lens, Euclidean GPU-style layouts impose congestion costs that can preserve near-quadratic pressure even when nominal attention complexity is reduced. As an illustrative topology-only proxy, a constrained optimization with $N = 128$ and $k \leq 4$ shifts from a planar baseline ($D = 21$) to an expander-like layout ($D = 5$), reducing hop-count routing proxy from $I_{irr}^{\text{hop}} = 2688$ to $I_{irr}^{\text{hop}} = 640$ (76%) and increasing proxy χ by 4.2× at fixed transceiver budget. Because this surrogate omits post-layout wire length, capacitance, and Rent-style pin-limited embedding effects, these gains are reported as upper bounds pending physical place-and-route validation.

1 Computation is Physical

For decades, theoretical computer science has relied on the foundational abstraction of the Turing machine. While this model provided the direct logical blueprint for the von Neumann architecture and modern digital chip design, its theoretical application often assumes an idealized physical environment: an infinite tape and state transitions that aren't grounded in physics. This abstraction of boundless memory and frictionless compute allowed algorithmic complexity to be evaluated independently of the underlying substrate. However, as artificial intelligence enters the regime of million-token context windows and trillion-parameter models, these idealized abstractions are increasingly intersecting with the hard physical constraints of modern silicon.

The emerging challenges in scaling foundational AI models are not strictly software limitations. They are fundamentally physical phenomena. Every matrix multiplication, attention score, and cached token demands physical instantiation. As Landauer's 1961 analysis made clear, the manipulation and erasure of bits are constrained by substrate physics; information processing is therefore inseparable from physical law [4, 2].

To design the next generation of computationally efficient machines and algorithms, we must expand upon purely dimensionless algorithmic analysis. The theoretical constructs of time and space complexity naturally map to the physical world. Time complexity serves as a proxy for irreversible information processing (I_{irr}), and space complexity a proxy for preserved information processing (I_{rev}). By analyzing algorithms not just as abstract mathematics, but in terms of physics, we establish a framework for evaluating computational efficiency and structural design in emerging

architectures.

2 Physical Foundations of Algorithmic Complexity

To operationalize algorithmic efficiency for physical hardware, we must translate the dimensionless abstractions of computational complexity into substrate-aware units that track conserved informational structure. We achieve this by defining computation not as a sequence of abstract mathematical steps, but as the active physical management of information. In a physical substrate, any algorithmic operation must fall into one of two categories: it either actively preserves a macroscopic physical state or irreversibly compresses it, a distinction developed in the Conservation-Congruent Encoding (CCE) framework (for a concise treatment, see *Toward a Physical Theory of Intelligence* [2]).

2.1 Mapping Time Complexity to Irreversible Information Processing (I_{irr})

In standard algorithmic analysis, time complexity ($\mathcal{O}(T)$) represents the number of sequential operations required to execute a function. However, on a physical substrate, standard logic operations (such as AND, OR, and standard memory overwrites) are logically irreversible. Two distinct input states converge into a single output state, meaning prior information is lost.

In thermal CMOS, Landauer's Principle gives one concrete lower bound: an erased bit requires at least $E = k_B T \ln 2$ of energy exchange, where k_B is the Boltzmann constant and T is ambient temperature. In CCE, this is a substrate-specific instantiation, not the universal unit of computational cost.

Consequently, the "time complexity" of a digital algorithm is a direct proxy for its irreversible physical burden. We formalize this physical cost as irreversible information processing (I_{irr}). Rather than measuring this burden in Joules, we abstract I_{irr} into fundamental natural units of information (*nats*). This keeps the metric substrate-agnostic: information may be encoded through different conserved quantities (for example charge, electromagnetic field structure, spin populations, or chemical potential), while I_{irr} still tracks the same irreversible compression of accessible state space.

2.2 Mapping Space Complexity to Preserved Structure (I_{rev})

Space complexity ($\mathcal{O}(S)$) traditionally measures the peak memory footprint required by an algorithm. In the context of scalable computational architectures, this manifests as the active structural state required to process, route, and maintain information over extended sequences.

Physically, it is a structural property maintained against ambient noise. To preserve information across time, the hardware must physically lock a substrate into a highly specific geometric configuration.

We define this preserved spatial structure as preserved information processing (I_{rev}). Unlike I_{irr}, which represents information permanently erased, I_{rev} represents the information actively sustained by the system's geometry.

3 Metric Definitions

The metric form developed in this section inherits its core structure from *Toward a Physical Theory of Intelligence* [2]: achieved causal work is normalized by physically grounded informational costs. In the present paper, however, several terms are specialized for computational-efficiency analysis. In particular, we map algorithmic time and space to irreversible information processing (I_{irr}) and

preserved information processing (I_{rev}), and use these quantities to evaluate architecture-level efficiency.

For the purpose of evaluating architectural efficiency, we separate task demand from architecture performance. We define target causal work (W_{target}) as the informational uncertainty that must be resolved for the task objective, and achieved causal work (W_{achieved}) as the uncertainty the architecture actually resolves at the required fidelity. Both are measured in *nats*, with $0 \leq W_{\text{achieved}} \leq W_{\text{target}}$.

When a target objective is known, the initial state possesses a mathematically quantifiable uncertainty. For example, an unsorted array of N elements has $N!$ possible permutations; predicting the next token in an AI model requires selecting from a vocabulary of size V. The hardware must do physical work to isolate the single, correct target state from the vast space of incorrect possibilities. Therefore, measured continuously in *nats*, W_{target} denotes the required informational volume of the task, while W_{achieved} denotes the resolved portion delivered by a specific architecture.

With this operational definition of algorithmic utility, we can define intelligence and structural consciousness as rigorous, dimensionless engineering constraints.

3.1 Intelligence (χ)

We define operational Intelligence (χ) as the ratio of achieved causal work to the irreversible information processing of the system:

$$\chi = \frac{W_{\text{achieved}}}{I_{irr}}$$

In this framework, χ serves as the ultimate metric for computational efficiency. It measures how effectively a hardware architecture converts physical phase space compression into useful task resolution. A brute-force algorithm scaling via redundant, unoptimized routing will generate a highly disproportionate I_{irr}, driving χ toward zero.

3.2 Consciousness (κ)

We operationalize Consciousness (κ) as a measure of structural capacity across physical space. We define κ as the ratio of achieved causal work to the preserved information processing:

$$\kappa = \frac{W_{\text{achieved}}}{I_{rev}}$$

Stripped of philosophical ambiguity, κ is a hard geometric metric. It measures how efficiently preserved structure in the system relates to causal work. A system with a low κ is structurally inefficient, requiring immense physical memory footprint to resolve minimal uncertainty. A system with a high κ achieves dense, highly compressed geometric representations, extracting maximum macroscopic causal utility from a minimal physical state.

3.3 Computational Efficiency

By isolating W_{achieved}, we reveal that intelligence and structural consciousness are mathematically bound by the underlying physical substrate. We can express the unified equation for computational efficiency as:

$$\chi = \kappa \left(\frac{I_{rev}}{I_{irr}} \right)$$

When output fidelity is relevant, we also report the retention ratio:

$$\rho = \frac{W_{\text{achieved}}}{W_{\text{target}}}, \quad 0 \leq \rho \leq 1$$

This unified ratio dictates that an architecture can only achieve high computational efficiency or intelligence (χ) if it possesses an optimized internal structure (κ), balanced perfectly against the ratio of its information processing. If a hardware substrate attempts to scale by simply inflating its active structural state (I_{rev}) without maintaining high structural density, the resulting irreversible routing costs (I_{irr}) will inevitably collapse the system's overall efficiency.

Accounting conventions for I_{rev} and I_{irr}. To keep cross-architecture comparisons reproducible, we use *preserved information processing* as the canonical meaning of I_{rev}. In this manuscript, I_{rev} includes all state that must remain physically distinguishable during inference at context length N (for example parameters, caches/recurrent state, and routing metadata). It excludes transient intermediates that are immediately discarded within a local operation. I_{irr} counts irreversible updates and communication events that erase prior distinguishability (for example overwrites and routing-induced state destruction). When retention depends on precision or noise floor, we make that dependence explicit in W_{achieved} rather than hiding it in asymptotic constants.

4 Analysis of Sorting Algorithms

To demonstrate the physical validity of these metrics, we apply them to a foundational pedagogical benchmark: sorting an array of N elements. In classical computer science, sorting algorithms are evaluated purely via their time and space complexities on digital hardware.

Before analyzing specific architectures, we first quantify the target causal work (W_{target}) required to solve the task. An unsorted array of length N possesses $N!$ possible permutations. The act of sorting collapses this uncertainty into a single, ordered target state. The total target work is therefore $\ln(N!)$ *nats*. Using Stirling's approximation for large N, we define:

$$W_{\text{target}} \approx N \ln N$$

This approximation assumes a maximum-entropy initial state over permutations. Adaptive algorithms that exploit prior order in the input correspond to a smaller effective W_{target} and therefore different I_{irr} profiles.

For exact sorting algorithms, we take $W_{\text{achieved}} = W_{\text{target}}$. To avoid invalid constant extraction from asymptotic notation, we report metric conclusions at the $\Theta(\cdot)$ level unless explicit constant-cost models are introduced. When a concrete retained-state multiplier is known, however, we keep that leading coefficient visible in I_{rev} rather than absorbing it into $\Theta(N)$, because it corresponds to real geometric burden on the substrate.

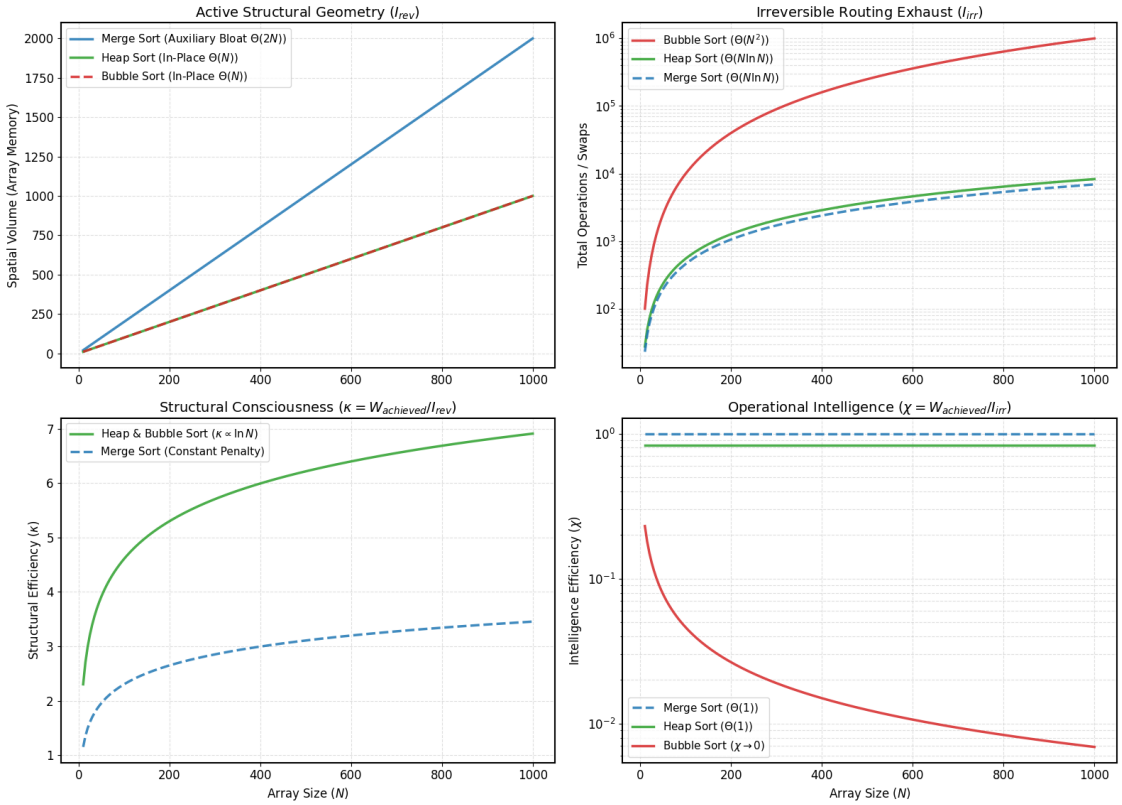

Figure 1: Scaling comparison of sorting algorithms under the proposed physical metrics, using $W_{\text{target}} \approx N \ln N$.

4.1 Bubble Sort

Bubble Sort operates in-place, requiring only the original array for memory, but relies on brute-force adjacent swapping to achieve order. For Bubble Sort, the corresponding scaling terms are:

$$I_{rev} = \Theta(N),$$
$$I_{irr} = \Theta(N^2),$$
$$W_{\text{achieved}} = \Theta(N \ln N).$$

Using our operational metrics:

$$\kappa = \Theta\left(\frac{W_{\text{achieved}}}{N}\right) = \Theta(\ln N)$$
$$\chi = \Theta\left(\frac{W_{\text{achieved}}}{N^2}\right) = \Theta\left(\frac{\ln N}{N}\right) \xrightarrow{N\to\infty} 0$$

Interpretation: Bubble Sort possesses high structural efficiency (consciousness, κ) because it does not bloat the spatial geometry of the hardware. However, at scale, the $\Theta(N^2)$ irreversible bit-flips and state erasures generate an irreversible routing burden that completely overwhelms the causal work achieved. The operational intelligence (χ) rapidly asymptotes to zero, representing a state of complexity saturation.

4.2 Merge Sort

Merge Sort uses a divide-and-conquer approach, minimizing operations but requiring an auxiliary array of comparable size to hold temporary states. For Merge Sort, the corresponding scaling terms are:

$$\begin{aligned} I_{rev} &\approx 2N \quad \text{(input array + auxiliary array)}, \\ I_{irr} &= \Theta(N \ln N), \\ W_{\text{achieved}} &\approx N \ln N. \end{aligned}$$

Using our operational metrics:

$$\kappa \approx \frac{W_{\text{achieved}}}{2N} \approx \frac{1}{2} \ln N$$

$$\chi = \Theta\left(\frac{W_{\text{achieved}}}{N \ln N} \right) = \Theta(1)$$

Interpretation: Merge Sort achieves constant-order operational intelligence ($\chi = \Theta(1)$), indicating near-lower-bound irreversible spend up to multiplicative constants. However, its explicit $2N$ retained-state burden halves κ relative to a strictly in-place design under the same target-work approximation.

4.3 Heap Sort

Heap Sort maintains a highly compressed implicit tree structure within the original array boundary while maintaining optimal operational efficiency. For Heap Sort, the corresponding scaling terms are:

$$\begin{aligned} I_{rev} &= N \quad \text{(strictly in-place)}, \\ I_{irr} &= \Theta(N \ln N), \\ W_{\text{achieved}} &\approx N \ln N. \end{aligned}$$

Using our operational metrics:

$$\kappa \approx \frac{W_{\text{achieved}}}{N} \approx \ln N$$

$$\chi = \Theta\left(\frac{W_{\text{achieved}}}{N \ln N} \right) = \Theta(1)$$

Interpretation: Heap Sort remains an efficient conserved-flow design: it attains constant-order operational intelligence while maintaining logarithmically growing structural density. Relative to Merge Sort, it avoids auxiliary geometric expansion and therefore retains the full leading coefficient in κ, making the physical advantage mathematically explicit rather than purely interpretive.

5 Analysis of Sequence Models

Having validated our operational metrics on classical sorting algorithms, we now apply them to a central architectural feature of modern artificial intelligence: autoregressive sequence modeling. We evaluate neural network architectures not just by their empirical benchmark scores, but by their fundamental efficiency.

To evaluate these architectures, we first define the target causal work (W_{target}) required to process and maintain a context window of N tokens. Unlike a sorting algorithm, which collapses the

uncertainty of $N!$ permutations within a closed set, sequence modeling must constrain a massive combinatorial state space. *A priori*, a sequence of N tokens drawn from a fixed vocabulary of size V presents V^N possible combinations. The maximum uncertainty reduction required to uniquely embed and hold this specific sequence is $\ln(V^N) = N \ln V$. Treating the vocabulary V as a constant, the baseline structural work scales as $W_{\text{target}} = \Theta(N)$. However, because natural language tokens are highly dependent, the architecture must not only hold these N states but actively route them to resolve their conditional relationships. Architectures are therefore compared by how efficiently they can preserve this target sequence (W_{achieved}) without being overwhelmed by the irreversible routing costs of discovering those dependencies.

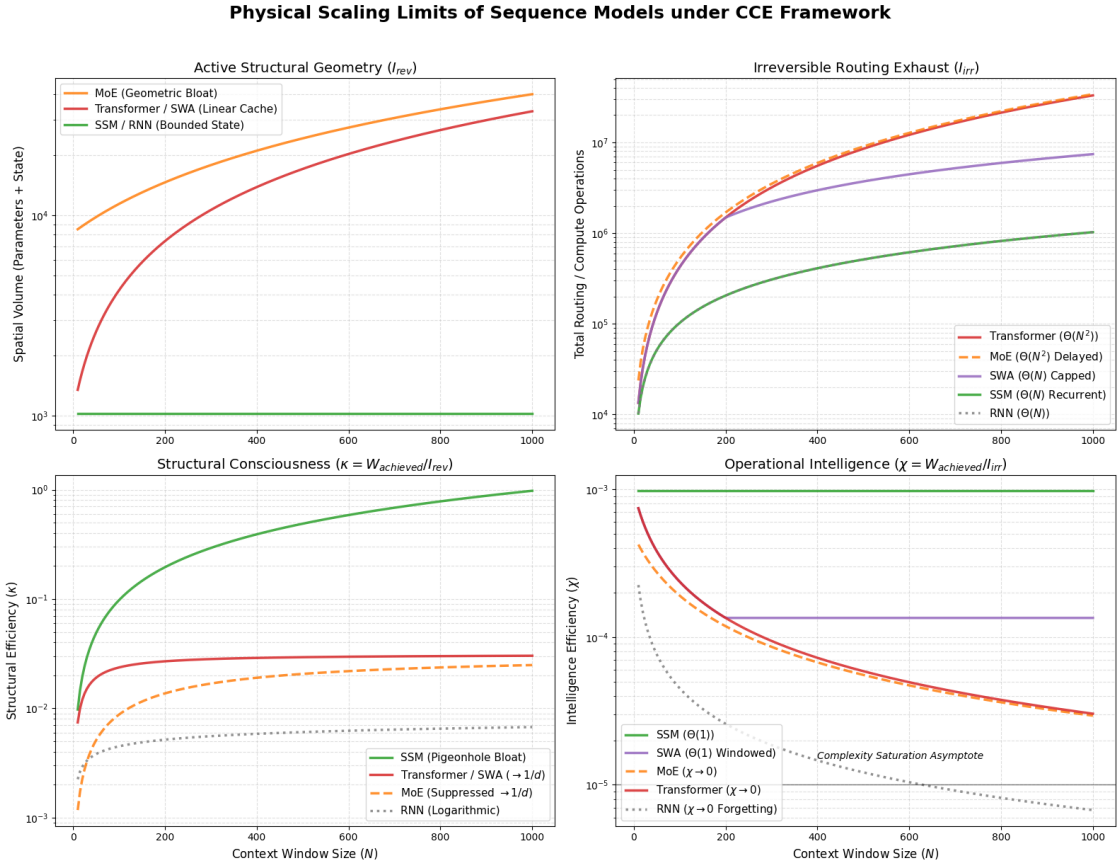

Figure 2: Scaling comparison of sequence-model architectures under the proposed physical metrics, using $W_{\text{target}} = \Theta(N)$.

5.1 Recurrent Neural Network (RNN)

The classic RNN processes sequences by compressing the entire past into a fixed-size hidden state vector of dimension d, updating it sequentially via dense matrix multiplication as new tokens arrive. While classical algorithmic analysis often treats model width d as constant relative to sequence length N, a physical conservation-law framework must account for both the geometry of the weight matrices and the irreversible burden of repeated state overwrites.

Crucially, while the target causal work to comprehend the sequence is $\Theta(N)$, the RNN cannot generally preserve this work losslessly. At each timestep, the historical state is subjected to a dense

affine transformation and a saturating nonlinearity. To prevent explosive instability, the eigenvalues of the transition matrix must be bounded, which induces an exponential decay of historical signal (the vanishing-gradient mechanism). With finite effective precision b, this decaying signal drops below resolvable amplitude after a finite horizon $h_{\text{eff}}(b)$.

For the standard RNN, the corresponding scaling terms are:

$$\begin{aligned} I_{rev} &= \Theta(d^2), \\ I_{irr} &= \Theta(N \cdot d^2), \\ W_{\text{achieved}} &= \mathcal{O}(\min\{N, h_{\text{eff}}(b)\}), \\ W_{\text{target}} &= \Theta(N). \end{aligned}$$

Here I_{rev} reflects the bounded static geometry of the $d \times d$ transition matrices, and I_{irr} captures the per-token dense matrix-vector updates accumulated across the full sequence. For fixed hardware precision, $h_{\text{eff}}(b)$ is independent of N.

Using our operational metrics:

$$\begin{aligned} \kappa &= \mathcal{O}\left(\frac{\min\{N, h_{\text{eff}}(b)\}}{d^2}\right) \\ \chi &= \mathcal{O}\left(\frac{\min\{N, h_{\text{eff}}(b)\}}{N \cdot d^2}\right) \xrightarrow{N \to \infty} 0 \\ \rho &= \mathcal{O}\left(\frac{\min\{N, h_{\text{eff}}(b)\}}{N}\right) \end{aligned}$$

Interpretation: The RNN's structural consciousness (κ) is bounded by the square of hidden dimension, so increasing capacity requires quadratically expanding static geometry (d). This raises irreversible routing exhaust (I_{irr}), while finite precision caps long-horizon retention through $h_{\text{eff}}(b)$. By continuously overwriting a fixed spatial volume, the architecture drives operational intelligence (χ) toward zero as context grows.

5.2 Baseline Transformer

To solve the RNN's forgetting problem, the Transformer architecture abandons the fixed hidden state and relies on global Attention [9]. While it shares a similar static footprint of $d \times d$ weight matrices for projections and feed-forward layers, its dynamic physical scaling fundamentally diverges from the RNN.

For the baseline Transformer, the corresponding scaling terms are:

$$\begin{aligned} I_{rev} &= \Theta(d^2 + N \cdot d), \\ I_{irr} &= \Theta(N \cdot d^2 + N^2 \cdot d), \\ W_{\text{achieved}} &= \Theta(N), \\ W_{\text{target}} &= \Theta(N). \end{aligned}$$

The I_{rev} term combines static parameter geometry with linear growth of Key-Value cache state. The I_{irr} term combines dense projection and feed-forward compute with quadratic global-attention routing. In this idealized accounting, the architecture preserves full-sequence target work, so W_{achieved} remains linear.

Using our operational metrics:

$$\kappa = \Theta\left(\frac{N}{d^2 + N \cdot d}\right) \xrightarrow{N\to\infty} \Theta\left(\frac{1}{d}\right)$$

$$\chi = \Theta\left(\frac{N}{N \cdot d^2 + N^2 \cdot d}\right) = \Theta\left(\frac{1}{d^2 + N \cdot d}\right) \xrightarrow{N\to\infty} 0$$

Interpretation: The Transformer solves the RNN's forgetting problem through brute-force global routing. Structurally, κ is capped at order $1/d$, forcing near one-to-one expansion of active memory with sequence length. Although retention can remain high ($\rho \approx 1$ under full-context attention), irreversible burden is dominated by the $N^2 \cdot d$ term. The architecture therefore trades width bottlenecks for temporal bottlenecks and trends toward complexity saturation at long contexts.

5.3 Mixture of Experts and Sliding Window Attention

To mitigate the compute costs of the Transformer, architectures often deploy algorithmic modifications. By evaluating these through the physical metrics introduced above, we can distinguish true structural cures from temporary scaling relief.

Mixture of Experts (MoE): MoE [7] attempts to reduce linear compute exhaust by expanding the static parameter footprint to E total experts, but only routing tokens to a sparse active subset (k). However, it retains the global quadratic Attention mechanism. For MoE, the corresponding scaling terms are:

$$\begin{aligned} I_{rev} &= \Theta(E \cdot d^2 + N \cdot d), \\ I_{irr} &= \Theta(N \cdot k \cdot d^2 + N^2 \cdot d), \\ W_{\text{achieved}} &= \Theta(N), \\ W_{\text{target}} &= \Theta(N). \end{aligned}$$

$$\kappa_{moe} = \Theta\left(\frac{N}{E \cdot d^2 + N \cdot d}\right) \xrightarrow{N\to\infty} \Theta\left(\frac{1}{d}\right)$$

$$\chi_{moe} = \Theta\left(\frac{N}{N \cdot k \cdot d^2 + N^2 \cdot d}\right) = \Theta\left(\frac{1}{k \cdot d^2 + N \cdot d}\right) \xrightarrow{N\to\infty} 0$$

Interpretation: MoE merely delays scaling collapse. It lowers the linear compute coefficient (k instead of E), but the $N^2 \cdot d$ routing burden still ultimately forces χ to zero at scale. Furthermore, its structural consciousness (κ) is heavily penalized at shorter context windows by the massive geometric bloat of storing inactive experts.

Sliding Window Attention (SWA): SWA [1] attempts to cap the quadratic explosion by strictly bounding the Attention mechanism to a local, fixed window of size W. For SWA, the corresponding scaling terms are:

$$\begin{aligned} I_{rev} &= \Theta(d^2 + N \cdot d), \\ I_{irr} &= \Theta(N \cdot d^2 + N \cdot W \cdot d), \\ W_{\text{achieved}} &= \Theta(N), \\ W_{\text{target}} &= \Theta(N). \end{aligned}$$

$$\kappa_{swa} = \Theta\left(\frac{N}{d^2 + N \cdot d}\right) \xrightarrow{N\to\infty} \Theta\left(\frac{1}{d}\right)$$

$$\chi_{swa} = \Theta\left(\frac{N}{N \cdot d^2 + N \cdot W \cdot d}\right) = \Theta\left(\frac{1}{d^2 + W \cdot d}\right) = \Theta(1)$$

Interpretation: SWA successfully stabilizes operational intelligence (χ) by physically bounding the irreversible routing burden to a constant relative to N. However, it operates at a permanently impaired baseline dictated by the window size ($W \cdot d$). Crucially, its structural consciousness remains stagnant; it does not physically compress the sequence geometry, leaving κ chained to the $1/d$ limit.

5.4 State Space Models (SSMs) and Linear Attention

Recent innovations, such as selective State Space Models (SSMs, e.g., Mamba [3]) and linear-attention architectures (e.g., RetNet [8]), attempt to map discrete sequence modeling to structured recurrent updates.

For SSMs and linear-attention models, the corresponding scaling terms are:

$$\begin{aligned} I_{rev} &= \Theta(d^2), \\ I_{irr} &= \Theta(N \cdot d^2), \\ W_{\text{achieved}} &= \mathcal{O}(\min\{N, h_{\text{eff}}^{\text{ssm}}(b,d)\}), \\ W_{\text{target}} &= \Theta(N). \end{aligned}$$

Unlike the Transformer's linear Key-Value memory growth, these architectures keep a bounded recurrent state that is independent of sequence length while preserving linear irreversible routing. Whether achieved causal work remains linear depends on precision, stability, and the effective retained horizon $h_{\text{eff}}^{\text{ssm}}(b,d)$.

Using our operational metrics:

$$\kappa = \mathcal{O}\left(\frac{\min\{N, h_{\text{eff}}^{\text{ssm}}(b,d)\}}{d^2}\right)$$

$$\chi = \mathcal{O}\left(\frac{\min\{N, h_{\text{eff}}^{\text{ssm}}(b,d)\}}{N \cdot d^2}\right)$$

Interpretation: Under idealized assumptions where $h_{\text{eff}}^{\text{ssm}}(b,d) = \Theta(N)$, these models remove the explicit quadratic term and recover constant-order $\chi = \Theta(1/d^2)$ with $\rho = \Theta(1)$. Under fixed finite precision, $h_{\text{eff}}^{\text{ssm}}$ may saturate below N, reducing retention and lowering practical efficiency. They therefore alleviate major scaling limits on current hardware, but do not universally eliminate them.

6 The Euclidean Substrate Diagnosis

The preceding analyses reveal a universal thermodynamic asymptote: on current hardware architectures, attempting to losslessly scale sequence comprehension inevitably triggers an explosion in irreversible routing exhaust (I_{irr}). To understand why algorithms like the Transformer undergo complexity saturation, we must abandon the assumption of a frictionless computational vacuum and examine the physical geometry of the hardware substrate itself.

The modern AI accelerator is fundamentally constrained by its 2-Dimensional Euclidean topology. While lithography nodes have shrunk to the nanometer scale, allowing for massive increases in transistor density, the macroscopic arrangement of Arithmetic Logic Units (ALUs) and High-Bandwidth Memory (HBM) remains a planar, Euclidean grid.

6.1 The Euclidean Congestion Penalty

In any 2D Euclidean space, the relationship between reachable volume and routing distance is strictly polynomial. If an operational node needs to communicate with its surroundings, the reachable area (and therefore memory capacity) expands quadratically with the routing radius (R), yielding $V \propto R^2$.

However, the communication bandwidth crossing any bisecting boundary of this space only scales linearly ($\propto R$). As the model demands larger contexts (increasing the required memory volume I_{rev}), the required routing paths necessarily bottleneck at the 2D perimeter. Therefore, any algorithm requiring global state comparison across the entire memory footprint—such as the Transformer's dense Attention mechanism—forces the physical hardware to shuttle massive blocks of data across increasingly long, highly congested, multi-hop physical wires. This wire-toggling manifests directly as the irreversible thermodynamic exhaust (I_{irr}) that destroys the system's operational intelligence (χ).

For dense all-pairs attention, this qualitative argument can be expressed as a bisection lower bound. Partition the N tokens into two equal halves. The number of cross-partition token interactions is at least

$$C_{\text{cross}} \geq \left(\frac{N}{2}\right)^2 d = \Omega(N^2 d),$$

where d is model width and each interaction carries $\Theta(d)$ state. Since each cross-partition interaction requires physical communication across the cut, irreversible routing burden inherits at least this scaling order under standard overwrite-based implementations:

$$I_{irr}^{\text{route}} = \Omega(N^2 d).$$

Additional hop distance and contention increase constants, but are not needed to recover the leading-order quadratic dependence.

6.2 The Geometric Mismatch of Sequence Data

This 2D hardware topology directly conflicts with the native geometry of the data it is attempting to process. Natural language and autoregressive sequences are not dense, flat matrices; they are inherently hierarchical. Syntactic structures, semantic contexts, and causal dependencies branch outward in a tree-like topology.

Mathematically, a hierarchical tree is a discretization of hyperbolic space. In a hyperbolic tree with a branching factor b and depth D, the total volume of reachable nodes expands exponentially: $V \propto b^D$.

The foundational crisis of modern AI hardware is a problem of topological embedding. We are attempting to embed an exponentially expanding, hyperbolic data structure onto a polynomially expanding, Euclidean silicon substrate.

6.3 The Embedding Collapse

To force a hyperbolic tree of size N into a 2D Euclidean grid, the physical distance between mathematically adjacent nodes must stretch exponentially. What should be a single structural hop ($\mathcal{O}(1)$) between a parent concept and a child token in hyperbolic space becomes an $\mathcal{O}(\sqrt{N})$ or even $\mathcal{O}(N)$ routing nightmare across the physical 2D silicon.

This geometric distortion helps explain the $\Theta(N^2)$ routing exhaust (I_{irr}) derived in Section 6.1. The Transformer's global attention matrix is not merely a software inefficiency; it is a substantial penalty of simulating hierarchical structure on a flat substrate.

Therefore, overcoming complexity saturation cannot be achieved by further sparsifying algorithms on planar chips. It physically necessitates a substrate that natively shares the topological congruence of the data—a non-Euclidean architecture.

7 The Hyperbolic Processing Unit (HPU)

Modern artificial intelligence is fundamentally bottlenecked not by logic gates, but by irreversible routing costs. The dominant architecture, the Transformer, relies on global Attention [9] to process sequences. To physically execute this on standard Graphics Processing Units (GPUs), the hardware must continuously broadcast and route token states across a massive 2D planar grid of Arithmetic Logic Units (ALUs) and memory buses. Because the spatial topology of a GPU is strictly Euclidean, performing an $\mathcal{O}(N^2)$ global state comparison requires shuttling data across the entire physical extent of the chip. Every movement and subsequent cache overwrite permanently erases previous states, generating immense irreversible routing exhaust (I_{irr}) in the substrate's conserved channels. As established in Section 6.2, this geometric mismatch between dense routing demand and flat 2D topology forces complexity saturation ($\chi \to 0$) as context windows scale. To improve scaling, we must align silicon architecture with non-Euclidean algorithmic geometry.

7.1 Theoretical Foundation: Hyperbolic Space

The core scaling failure of the Transformer is that it treats temporal sequences as flat, linear arrays; to retrieve a specific context, it must route across the entire array.

Hyperbolic geometry, conversely, is characterized by exponential expansion of spatial volume as a function of radius (visualized in 2D as the Poincaré disk). This expansion mirrors tree-like hierarchies [6]. Under low-distortion embedding assumptions, hierarchical sequence dependencies can be represented with depth $\mathcal{O}(\ln N)$ rather than requiring dense all-pairs routing. Retrieval can then be approximated by sparse directed path traversal instead of dense $\mathcal{O}(N^2)$ matrix search.

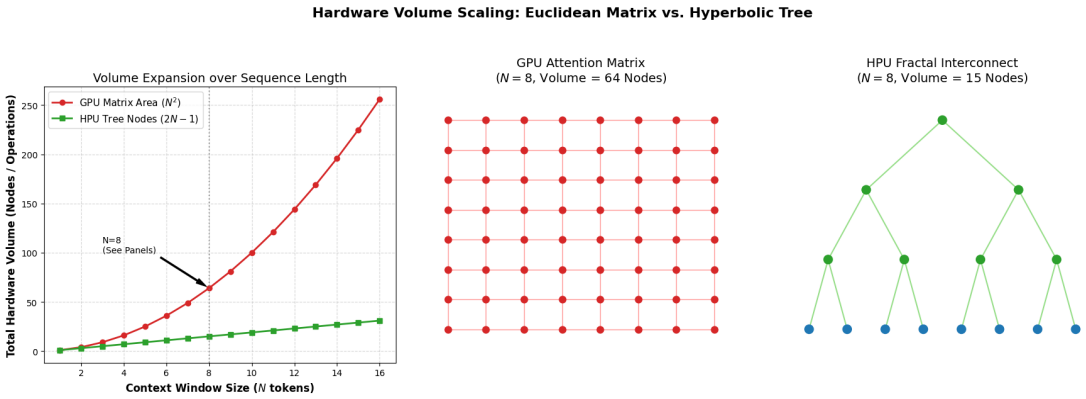

Figure 3: Illustrative hyperbolic geometry (Poincaré-disk view), showing how spatial capacity expands with radius and supports hierarchical routing.

7.2 Physical Architecture of the HPU

A standard GPU separates a grid of compute (ALUs) from a pool of memory (VRAM), communicating across a massive, power-hungry data bus. The HPU abandons this von Neumann topology entirely, substituting dense linear algebra for *Conditional Geometric Compute*.

The Interconnect (Fractal Fat-Trees): Instead of a 2D "Manhattan street grid" Network-on-Chip (NoC), the HPU's physical silicon traces a fractal, branching topology. Data enters a root node and routes strictly outward down the branches. To circumvent planar layout limits, the HPU relies heavily on 3D-stacked Through-Silicon Vias (TSVs) to build an expanding physical pyramid.

The Memory (Colocated Hierarchical SRAM): There is no central pool of VRAM. Instead, memory is physically distributed across the branches of the interconnect tree. When a signal routes down a specific branch, the relevant state context is physically millimeters away from the routing logic.

The Compute (Hyperbolic ALUs): The HPU replaces standard Multiply-Accumulate (MAC) units with custom ALUs engineered natively for hyperbolic distance metrics. Instead of summing vectors, a node calculates the distance between the incoming token and its downstream branches in hyperbolic space, acting as a physical, probabilistic router.

7.3 Physical Evaluation under the Proposed Metrics

We evaluate the physical viability of the HPU using our operational metrics, fixing sequence-level target work at $W_{\text{target}} = \Theta(N)$ and assuming idealized full retention ($W_{\text{achieved}} = \Theta(N)$). To keep this assumption internally consistent, we decompose preserved information into a retained-state term and a routing-metadata term:

Retained State (I_{state}): $\Theta(N)$

Routing Metadata (I_{route}): $\Theta(\ln N)$

Preserved Information Processing (I_{rev}): $I_{\text{state}} + I_{\text{route}} = \Theta(N + \ln N) = \Theta(N)$

Irreversible Exhaust (I_{irr}): $\Theta(N \ln N)$

Plugging these physical costs into our unified equations yields a routing advantage without requiring contradictory memory scaling assumptions.

Operational Intelligence (χ):

$$\chi_{HPU} = \Theta\left(\frac{W_{\text{achieved}}}{I_{irr}}\right) = \Theta\left(\frac{1}{\ln N}\right)$$

Result: Unlike the Transformer, which degrades at order $1/N$, the HPU decays at order $1/\ln N$. This is still asymptotically decreasing, but materially slower over practical context sizes.

Structural Consciousness (κ):

$$\kappa_{HPU} = \Theta\left(\frac{W_{\text{achieved}}}{I_{rev}}\right) = \Theta\left(\frac{N}{N + \ln N}\right) = \Theta(1)$$

Result: In this accounting, the architecture retains constant-order structural efficiency while improving routing efficiency. Because retained state contributes $\Theta(N)$ by definition, any derivation

of $\kappa_{HPU} = \Theta(N/\ln N)$ is inconsistent with the metric decomposition and corresponds to dropping the dominant state-retention term from I_{rev}.

7.4 Sparse Backpropagation: The Learning Efficiency Advantage

The ultimate advantage of the HPU emerges during training. On a GPU, calculating gradients via backpropagation requires activating the entire 2D grid simultaneously to update every matrix weight.

In the intended HPU design, a forward pass activates only a single $\mathcal{O}(\ln N)$ path through the fractal tree. Causal credit assignment is therefore localized to the same routed path. In that regime, backpropagation becomes a sparse geometric adjustment rather than a global dense-update event.

7.5 Empirical Validation via Topological Optimization

To convert this analysis from a descriptive tool to a prescriptive engineering framework, we first distinguish graph-level hop reduction from post-layout physical reduction. With communication events m routed along edge paths $P(m)$, an embedding-aware routing burden is

$$I_{irr}^{\text{route,phys}} \propto \sum_{m} \sum_{e \in P(m)} \lambda_e,$$

where λ_e absorbs wire-length-, capacitance-, and congestion-dependent toggle burden (for example $\propto C_e V^2$). In practical VLSI, λ_e is strongly constrained by placement and pin-limited fan-out (Rent-style scaling), so topological adjacency alone does not determine physical cost.

Because full placement-and-routing data are not yet available for this conceptual HPU, we use a transparent surrogate objective based on unembedded hop count,

$$I_{irr}^{\text{hop}} \propto \sum_{m} |P(m)|,$$

and treat improvements in this quantity as upper-bound indicators rather than direct thermodynamic measurements.

Under this surrogate, we map chiplets to graph vertices ($|V| = N$) and interconnect candidates to edges, and minimize routing diameter D subject to a local degree budget k (a routing-fabric component of I_{rev}, not total preserved state).

Mathematically, this maps the physical scaling laws of AI directly onto the degree-diameter problem and its Moore-style upper bound [5]:

$$N \leq 1 + k \sum_{i=1}^{D} (k-1)^{i-1}$$

To empirically validate the topological component of this claim, we simulated a localized network of $N = 128$ compute nodes, restricted to a strict geometric limit of $k = 4$ wires per node.

For reproducibility, the illustrative optimizer is defined explicitly: objective $\min D$ (proxy for minimizing I_{irr}^{hop}), constraint set ($N = 128$, $k \leq 4$) with graph connectivity preserved, initialization from the planar baseline, and iterative single-edge swaps accepted only when they improve D. This is sufficient to replicate the reported trajectory qualitatively; full physical claims still require post-layout extraction and run logs.

The Euclidean Baseline (Hop-Count Proxy): Mapping the nodes to a standard 2D planar grid resulted in a routing diameter of $D = 21$, corresponding to $I_{irr}^{\text{hop}} = 2688$ in the unembedded surrogate.

The Expander Candidate (Same Proxy): We deployed a heuristic edge-swapping topology optimizer governed strictly by the surrogate objective (minimizing I_{irr}^{hop} without violating $k \leq 4$). The optimizer rejected planar layouts and converged to an expander-like topology.

Results: Under identical N, k constraints, the expander-like candidate compressed diameter from 21 hops to $D = 5$ (one hop above the Moore-style benchmark $D = 4$).

In surrogate routing terms, this dropped I_{irr}^{hop} to 640 (a 76% reduction). Consequently, proxy operational intelligence increased by 320% (4.2×, or to 420% of baseline). Crucially, this is not yet a post-layout energy claim: long global links in 2D/3D embedding can carry much larger physical cost than local links, so validation requires embedding-aware place-and-route with extracted wire lengths and RC/toggle models.

8 Conclusion

For decades, theoretical computer science and hardware engineering have operated across an artificial divide: algorithms were treated as dimensionless abstractions, while hardware was treated as a passive substrate. The Conservation-Congruent Encoding (CCE) framework developed in this paper closes this gap by mapping algorithmic time and space complexity directly onto irreversible processing burden (I_{irr}) and preserved structural geometry (I_{rev}).

Under this lens, the current AI scaling crisis is fundamentally physical. The dominant Transformer architecture is not stalling because of insufficient data or software ingenuity; it is approaching complexity saturation due to quadratic routing costs. Techniques such as MoE mainly delay this asymptote. SSM and linear-attention families can remove the explicit quadratic term in idealized models, but practical gains remain bounded by finite-state precision and substrate constraints on legacy hardware.

The proposed Hyperbolic Processing Unit (HPU) provides a candidate path beyond this limit by aligning silicon architecture with the non-Euclidean geometry of hierarchical sequence data. By utilizing a fractal, tree-routed, memory-colocated substrate, the HPU aims to alter scaling laws of retrieval and backpropagation, potentially avoiding the worst effects of $\mathcal{O}(N^2)$ routing collapse.

Realizing these theoretical gains will require crossing significant engineering thresholds and publishing reproducible hardware-software evidence. In particular, the $N = 128$ expander result should be interpreted as a topology-level upper bound, not a fabricated-chip energy result; embedding-aware validation must include wire length, congestion, clocking, and Rent-style pin-scaling constraints. The CCE viewpoint indicates that the next era of AI progress will likely come not from larger planar compute grids or software tricks alone, but from co-designing algorithms and hardware around conservation laws and geometric congruence, while allowing information to be encoded in whichever conserved quantities a substrate supports.

References

[1] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The Long-Document Transformer. arXiv preprint arXiv:2004.05150, 2020.

[2] Peter David Fagan. Toward a Physical Theory of Intelligence. arXiv preprint arXiv:2601.00023, 2026.

[3] Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv preprint arXiv:2312.00752, 2023.

[4] Rolf Landauer. Irreversibility and Heat Generation in the Computing Process. *IBM Journal of Research and Development*, 5(3):183–191, 1961.

[5] Mirka Miller and Jozef Širáň. Moore Graphs and Beyond: A Survey of the Degree/Diameter Problem. *Electronic Journal of Combinatorics*, DS14, 2013.

[6] Maximilian Nickel and Douwe Kiela. Poincaré Embeddings for Learning Hierarchical Representations. In *Advances in Neural Information Processing Systems 30*, pages 6338–6347, 2017.

[7] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey Hinton, and Jeff Dean. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *International Conference on Learning Representations*, 2017.

[8] Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive Network: A Successor to Transformer for Large Language Models. arXiv preprint arXiv:2307.08621, 2023.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008, 2017.