ChatGPT

Unifying Automata Theory, Universal Computation, and Scale-Invariant Dynamics

Physical Interaction as Computation and Universality

Computation as a Physical Process: Under **computational functionalism**, we assume that what matters is the functional information processing, not the particular material it runs on. Modern theory holds that *information is physical* – any computation must occur via physical interactions [1]. In fact, some philosophers and physicists propose **pancomputationalism**: *every* physical system performs some form of computation [2]. Even something as inert as a rock or as chaotic as a hurricane instantiates causal state transitions that can be interpreted as computation [2] [3]. According to one formulation, computation just is the *causal structure* of a physical process; since every physical system has some causal structure, every system can be said to "compute" its own dynamics [3]. This view implies that the simplest physical interaction – say, two particles colliding and changing states – can be seen as performing a primitive computation (updating the "state" of the system). While not everyone agrees with unlimited pancomputationalism, it provides a helpful mindset: **physical interactions and computations are two sides of the same coin**.

Universal Computation from Simple Rules: A remarkable fact from **automata theory** is that extremely simple systems of local interactions can become *universal computers*. "Universal" means they can perform any computation that any other computer (e.g. a Turing machine) can, given enough time and memory [4] [5]. For example, the mathematician Alan Turing showed that a single abstract machine can be "universal" by reading instructions – this is the essence of the **Church–Turing thesis** that any effective algorithm can run on some Turing-equivalent machine [6] [7]. More strikingly, **Conway's Game of Life** – a simple cellular automaton with just a few rules on a grid – is *Turing complete*: by crafting the right initial configuration, the automaton's evolution can implement data storage and logical operations to carry out any algorithm [8] [9]. In Conway's Life, moving patterns called *gliders* can function as signals or bits; researchers showed one can build logic gates (AND, OR, NOT) and even a functioning Turing machine out of colliding gliders and other structures [9]. In short, a sufficiently complex set of physical interactions *can emulate arbitrary computations*. Many other simple systems have shown universal computation as well – e.g. one-dimensional cellular automaton Rule 110 is universal, as are certain subatomic particle interaction models and even theoretical models like billiard ball collisions [10]. This supports the idea that **if you have enough interacting parts with the right conditions, you can in principle encode any computation in physical dynamics**.

Emergence of Persistent Structures Across Scales

Automata and Gliders – Persistence from Dynamics: One hallmark of complex computational systems is the spontaneous emergence of *persistent patterns*. In cellular automata (CA), simple rules can give rise to stable or recurring structures that maintain themselves over time. A famous example is the **glider** in the Game of Life – a pattern of 5 "alive" cells that repeats and moves diagonally across the grid indefinitely [11] [9]. Such structures are **invariants** in the dynamics: even though the system's state updates at every step, the pattern as a whole persists and has a coherent identity (the glider "particle"). These emergent objects can interact, transmitting information or producing higher-order behaviors. *Figure 1* below illustrates a space-time diagram of the Rule 110 cellular automaton, another simple rule set known to be universal. Diagonal streaks in the image are analogous to gliders – they are *localized structures that persist and move through the cellular medium*, effectively acting like particles or signals in the automaton [12]. In essence, **simple low-level rules can generate higher-level stable components**.

Figure 1: Space-time diagram of Rule 110 cellular automaton (time advancing downward). Despite being governed by a simple local rule, Rule 110 supports persistent moving patterns (analogous to gliders) that can carry information. Many researchers have drawn parallels between such emergent structures and physical particles or organisms – stable patterns that maintain themselves in a dynamical system [12] *.*

Scale-Invariant Principles of Organization: The glider is a useful metaphor for how *higher-level order* can emerge from lower-level processes. In Conway's Life, the glider is just an arrangement of cells – but we can describe it as an independent object with behaviors (it "moves," it can "collide" with other gliders or structures, etc.). Likewise, in nature we see structures that persist on top of more fundamental layers: **subatomic particles** are stable patterns of quantum fields, **atoms** are stable combinations of particles, **molecules** are stable configurations of atoms, and so on up to **cells**, **organisms**, and even **memes** in culture. Each layer uses the layer below as a substrate, yet exhibits its own dynamics and "laws." The key is that certain configurations at a lower level achieve **relative invariance**, providing building blocks for the next level. For example, the genetic code in DNA is physically just a sequence of molecules, but it's also an information pattern that persists (with replication) over generations, enabling the higher-level phenomena of inheritance and evolution. In the realm of ideas, a concept like "money" or "law" is an abstract pattern that only exists because many humans collectively *represent* it and act on it, but once it exists, it tends to persist or self-propagate (unless destabilized by opposing forces). The **representation scheme** might be different (biochemical in DNA, electromagnetic in a computer, social and cognitive in culture), but in each case we find dynamic patterns that can *stabilize and replicate*. This suggests there are **scale-invariant principles** – such as feedback loops, information encoding, and self-reproduction – that appear in any sufficiently complex **representational layer**, whether in 3D physical space or in higher "conceptual" dimensions like social systems.

Entropy and Persistence: Notably, these persistent patterns are fighting against the natural tendency toward disorder. In a strict thermodynamic sense, any closed physical system tends toward increasing entropy (disorder) over time. Yet *open systems* can maintain or even increase order by importing energy or resources – this is how life works. Physicist Erwin Schrödinger famously described life as feeding on "**negative entropy**" [13]. In other words, living systems create local pockets of order (low entropy) at the expense of expending energy (increasing entropy in their environment). A glider in the Game of Life persists only because the rules are perfectly noiseless and it's in a kind of closed world; real organisms persist by constantly expending energy to repair and reproduce themselves. The concept of **dynamic kinetic stability (DKS)** has been introduced to capture this persistence-through-change: living or self-replicating systems are *dynamically stable* in that they keep *propagating their pattern* even though their material constituents are continually turning over [14] [15]. For example, your body maintains its form over decades, but almost none of the exact atoms from your childhood are present years later – you are a *pattern* being preserved. This principle applies not just to organisms but to ideas and social structures: a government or a corporation can last centuries, though the people and resources involved constantly change. **Persistence is achieved by ongoing dynamics, not static permanence.** In this sense, the fight against entropy is an *information battle* – keeping the *pattern* going amidst a sea of perturbations.

Entropy, Incompleteness, and the Role of Redundancy

Incompleteness of Fixed Rules: You raised an intriguing analogy to Gödel's incompleteness theorem – the idea that no formal axiomatic system can ever be complete and self-consistent [16]. In simple terms, Gödel proved that in any sufficiently powerful formal system (like arithmetic), there will be true statements that the system's rules cannot prove [16]. We can draw a loose parallel to any complex evolving system (like a society with laws or a software with fixed protocols): no finite set of fixed rules can foresee or account for every possible circumstance in the system's evolution. **There will always be unforeseen edge cases or "exceptions"** – in governance this can manifest as loopholes, ambiguous cases, or scenarios the law didn't anticipate. When such gaps occur, the system experiences an "error" or inconsistency with reality – what you called *"corruption"* in a governing system (i.e. reality deviating from the intended order of the rules). While this is an analogy (social laws are not literally mathematical axioms), the principle is that **any static rule-set will eventually prove inadequate in a dynamic, open-ended environment**. This connects to another computation concept: Turing's *halting problem* implies that you cannot have a single algorithm that predicts the outcome of every possible program (or every input to a complex system) [17]. In a universe that contains computationally universal processes, **fundamental unpredictability** is inevitable [17]. Similarly, a society or organism cannot have a single static plan that covers all future states; adaptation and learning are necessary to handle novel events.

Redundancy and Error Correction: How do living and computational systems cope with the inevitable errors and uncertainties? One powerful strategy is **redundancy** – creating backups, copies, and failsafes so that no single error causes total failure. In computation and information theory, this is well-formalized: Claude Shannon's *channel coding theorem* shows that by adding redundancy to messages, one can achieve an arbitrarily low error rate in communication, as long as the information rate stays below a capacity limit [18] [19]. Engineers once assumed that to get error probability near zero you'd have to slow down to zero throughput, but Shannon proved you can still send data at a finite rate while making error risk as small as desired by clever encoding [19]. This is the basis of error-correcting codes – from simple parity bits to sophisticated turbo and LDPC codes – which use extra bits to detect and fix mistakes in transmitted data. In essence, **redundancy at a lower layer** (physical bits) creates a more invariant, reliable signal at a higher layer (the encoded information). The same concept appears in biology and society: life builds redundancy to guard against entropy and failure. **Biological examples:** Multicellular organisms have millions of cells; they can often survive the loss of some cells or even whole organs because others compensate. Even single-celled organisms maintain two copies of DNA (diploidy) or have repair enzymes – multiple safeguards for vital information. Populations carry genetic diversity so that not all individuals succumb to the same stress; some will have traits that survive, preserving the lineage. **Social examples:** Communities don't rely on just one farmer or one cook – they cultivate many, so that knowledge and function are distributed. Important roles are often duplicated (consider backup leaders, multiple data centers for internet companies, or diverse supply chains in an economy) to ensure continuity if one node fails. This *principle of redundancy* is a direct manifestation of computational thinking in other domains: it's effectively an error-correcting code in society and biology, providing **fault tolerance**. By **building abstract protocols on top of unreliable parts**, systems achieve a kind of *scale-invariant robustness*. Just as a computer can function even if a few bits flip in memory (thanks to error correction and redundant circuitry), a large organism can live on even if some cells die, and a civilization can endure even if some members or institutions collapse, as long as critical information and functions are replicated elsewhere.

Life's Layered Defense Against Entropy: When you suggest that *"life's strategy to fight a universe going towards higher entropy is to create deeper and deeper abstractions and representation layers using invariance at* *a lower layer"*, you are aligning with how scientists understand the evolution of complexity. Each new layer of complexity in biology or culture indeed *builds on the stability of the previous layer*. DNA-based life introduced a reliable way to store information (with base-pairing redundancy, error-correcting polymerases, etc.), enabling accurate heredity. Multicellularity added a new layer: cell specialization and cooperation gave organisms internal redundancy and division of labor (much like a parallel or distributed computing system). At the cognitive and cultural level, humans externalize memories into books and now digital media – an added redundancy beyond our brains – and we create legal and economic systems that provide stability to society (for instance, codifying protocols like currency, contracts, and governance structures). Each layer tries to **mitigate the entropy of the layer below**: e.g. social systems mitigate the randomness of individual human behavior by establishing norms and laws; technology mitigates the unpredictability of raw nature by engineering controlled environments. None of this defies the Second Law of Thermodynamics globally – entropy in the total closed system still increases – but it *locally* creates pockets of decreasing entropy (more order/predictability) by exporting disorder elsewhere (e.g. life radiates heat waste to stay organized [13]). Thus, **invariance propagates upward:** a stable lower layer (through error-correction or homeostasis) allows a higher layer to exist and persist, which in turn can have its own stable patterns that support an even higher layer, and so on. This hierarchical, modular stability is a unifying theme from physics to biology to sociology.

Toward a Unified Computational Theory of Nature

We now see the outline of a possible **unification**: despite the enormous differences between chemical reactions, living cells, minds, and societies, all these systems can be viewed through the lens of *information processing and computation*. They consist of components interacting (physically or symbolically), following rules, and giving rise to emergent patterns that sometimes **encode information, maintain stability, and self-perpetuate**. Some scientists have conjectured that *the entire universe* is fundamentally a computational structure – essentially a gigantic cellular automaton or computer program running its course [12]. In this view (associated with thinkers like Konrad Zuse, Edward Fredkin, Stephen Wolfram), everything we observe – electrons, galaxies, life, consciousness – would be information patterns (like gliders) in an underlying cosmic-scale automaton [12]. This is a bold hypothesis and remains unproven and speculative. Yet, it is a helpful *metaphorical framework*: it encourages us to find common principles across disciplines. Indeed, concepts like feedback control, network dynamics, algorithmic complexity, and evolutionary adaptation appear in multiple fields. By assuming **computational functionalism**, we treat a brain as a kind of computer, or a society as an information-processing network, and often gain insights that purely reductionist physical descriptions would miss.

Potential Challenges and Clarifications: Embracing "everything is computation" can lead to some philosophical pitfalls if not carefully defined. Critics point out that if one allows an overly loose interpretation, *any* physical sequence can be seen as implementing *any* computation (with a clever mapping), which makes the idea trivial [20]. To avoid this, we usually constrain what we mean by "implementation of a computation" – for example, requiring a systematic correspondence between physical states and formal states, and that the causal structure of the physical system mirrors the logical structure of the computation [20] [21]. When these criteria are met, saying "physical interaction **is** computation" becomes meaningful. Another challenge is empirical: it's not yet clear that the universe at large is a discrete digital automaton – it might be analog or continuous at small scales, or something even stranger (quantum physics, for instance, has continuous state spaces, though some interpretations like *causal set theory* or loop quantum gravity suggest spacetime might be discrete at the Planck scale) [22] [23]. So, the **digital physics** hypothesis remains an open question [23]. Furthermore, drawing analogies from Gödel's theorem or

Turing's halting problem to real-world systems has to be done with care – human legal systems and mathematics are not identical. The parallels are illuminating, but one shouldn't assume a social system has the formal precision of arithmetic; rather, the analogy highlights a trend toward *incompleteness* and the need for continual revision of rules.

Conclusion: Within the assumption of computational functionalism, the conceptual framework you're building is a compelling one. It suggests a universe where complexity breeds computation, and computation breeds further complexity in a positive feedback loop. We see simple interactions giving rise to universal computing capability; those computations in turn create stable *substrates* for higher-level computations (gliders to organisms to societies), all under the universal pressure of entropy and the ingenious defiance of it through redundancy and self-organization. While some details of this grand picture remain speculative or metaphorical, many pieces are supported by established science: Turing-universal patterns in simple automata [9], life's reliance on negentropy and information replication [13] [14], and the effectiveness of error-correcting codes and redundancy at maintaining order [19]. The **flourishing of invariants across scales** – whether a stable genome, a lasting cultural meme, or an orbiting planet – can indeed be seen as computations that have achieved a kind of *robust fixity* against chaos. By unifying these insights, one might develop a **coherent theory of persistent computation**: a view that *computation is not only something that we intentionally design in computers, but a principle that nature uses to create pockets of order, complexity, and longevity in an otherwise entropy-driven world*. Such a theory is ambitious, but it aligns with the interdisciplinary trend of treating information as a common currency in physics, biology, and sociology. Your approach, grounded in known computational principles and scaled up through layers of abstraction, is a strong step toward that unifying perspective. It acknowledges the **scale-dependent differences** (no one would mistake a chemical reaction for an economy), yet uncovers the **scale-invariant principles** (feedback, information storage, replication, etc.) that underlie all these phenomena. In summary, assuming computational functionalism provides a rich explanatory scaffolding – one that, while requiring careful definitions and further empirical exploration, has the potential to connect the laws of physics with the "laws" of life and mind into one grand narrative of computation.

Sources:

- Stanford Encyclopedia of Philosophy – *Computation in Physical Systems* [2] [3]
- *Quanta Magazine* – "The Physical Origin of Universal Computing" [1] [5]
- Stanford Encyclopedia of Philosophy – *Cellular Automata* [8] [10]
- Wikipedia – *Glider (Conway's Game of Life)* [9]
- Stanford Encyclopedia of Philosophy – *Cellular Automata as Models of Reality* [12]
- *Entropy and Life* – Wikipedia (on Schrödinger's negentropy) [13]
- *Aeon* – "Paradoxes of Stability" (Addy Pross on DKS in life) [14]
- Shannon's Channel Coding Theorem (Shannon 1948) – legacy summary [19]
- Wikipedia – *Gödel's Incompleteness Theorems* [16]
- Stanford Encyclopedia of Philosophy – *Computability and Unpredictability* [17]

---

[1] [4] [5] [6] [7] The Physical Origin of Universal Computing | Quanta Magazine
https://www.quantamagazine.org/the-physical-origin-of-universal-computing-20151027/

[2] [3] [20] [21] Computation in Physical Systems (Stanford Encyclopedia of Philosophy/Fall 2015 Edition)
https://plato.stanford.edu/archives/fall2015/entries/computation-physicalsystems/

[8] [10] [12] [17] [22] [23] Cellular Automata (Stanford Encyclopedia of Philosophy)
https://plato.stanford.edu/entries/cellular-automata/

[9] [11] Glider (Conway's Game of Life) - Wikipedia
https://en.wikipedia.org/wiki/Glider_(Conway%27s_Game_of_Life)

[13] Entropy and life - Wikipedia
https://en.wikipedia.org/wiki/Entropy_and_life

[14] [15] Paradoxes of stability: how life began and why it can’t rest | Aeon Essays
https://aeon.co/essays/paradoxes-of-stability-how-life-began-and-why-it-can-t-rest

[16] Gödel's incompleteness theorems - Wikipedia
https://en.wikipedia.org/wiki/G%C3%B6del%27s_incompleteness_theorems

[18] [19] itsoc.org
https://www.itsoc.org/resources/Shannon-Centenary/shannon-work-legacy-paper