

# Computation as Semantic Maintenance: Embodied Knowledge, Structural Capability, and the Laws of Development

Flyxion

December 28, 2025

## Abstract

This essay develops a unified theory of knowledge grounded in embodiment, coordination, and irreversibility, building on the empirical laws articulated in César Hidalgo's *The Infinite Alphabet and the Loss of Knowledge*. Hidalgo argues that knowledge is non-fungible, difficult to diffuse, and unevenly distributed across societies due to deep structural constraints. Here these insights are elevated from empirical regularities to a coherent ontology of collective capability.

Knowledge is treated not as a substance that accumulates, nor as information that transfers intact, but as a property of organized systems sustained through ongoing coordination among people, machines, and institutions. Growth is shown to depend on architectural stabilization rather than individual learning alone. Diffusion is reconceptualized as local reconstruction under constraint rather than spatial transmission. Valuation is revealed to be necessarily indirect, measuring the shadow cast by embodied capabilities rather than knowledge itself.

The essay then extends this framework by introducing an event-historical account of collective cognition, formalized through Spherepop semantics, and by deriving Hidalgo's laws from a relational field perspective inspired by scalar, vector, and entropic dynamics. The result is a theory in which economic development is understood as the maintenance of coherence in complex systems, and knowledge loss as the default outcome when structural alignment fails. Policy implications are reframed accordingly, shifting emphasis from accumulation to stewardship of semantic and organizational infrastructure.

## 1 Knowledge as a Property of Organized Matter

The persistent difficulty of growing, moving, and valuing knowledge has long tempted economists and policymakers to treat it as an elusive substance: a resource that behaves unlike capital, labor, or energy, yet is assumed to circulate, accumulate, and depreciate according to analogous principles. César Hidalgo's *The Infinite Alphabet and the Loss of Knowledge* decisively rejects this intuition by insisting that knowledge is not a thing at all. It is a property.

The analogy Hidalgo offers is precise rather than metaphorical. Temperature is not a particle that flows between objects; it is a macroscopic property emerging from the kinetic organization of matter. No amount of description or archival record can store temperature independently of the system that instantiates it. Likewise, knowledge is a macroscopic property of organized physical

systems, arising from the structured coordination of people, tools, machines, and social roles. It has no independent existence apart from those arrangements.

This ontological correction has immediate and far-reaching consequences. Books, manuals, blueprints, and digital files do not contain knowledge. They are records that may support the reconstruction of knowledge, but only if the physical and social systems capable of enacting that reconstruction remain intact. When those systems dissolve, knowledge does not wait patiently on the page; it disappears. The loss of knowledge is therefore not primarily a failure of memory or storage, but a failure of organization.

Treating knowledge as a property rather than a substance dissolves many long-standing confusions. It explains why knowledge is difficult to transfer, why it resists simple aggregation, and why it decays even in the presence of perfect archives. It also reveals why economic development is inseparable from the physical and institutional architectures that sustain coordinated action. Knowledge cannot be imported like raw materials or accumulated like currency. It must be continuously enacted by systems capable of holding it together.

This view immediately reframes the problem of development. If knowledge is a property of organized matter, then development is not the accumulation of abstract inputs such as education, information, or technology in isolation. It is the process by which societies build, stabilize, and repair the structures that allow complex capabilities to remain embodied over time. Growth, diffusion, and valuation must therefore be understood as consequences of structural coherence rather than as independent economic processes.

## 2 Embodied Knowledge in Ordinary Practice

The claim that knowledge is a property of organized systems rather than a transferable substance can appear abstract unless it is anchored in ordinary forms of skilled practice. Everyday domains such as cooking, carpentry, electrical work, plumbing, and childcare provide especially clear examples, precisely because they resist formalization while remaining indispensable.

Consider cooking. A recipe does not contain the knowledge required to produce a meal. It records a sequence of symbolic instructions, but the successful execution of those instructions depends on embodied judgments about texture, timing, heat, and substitution that are acquired only through practice. Two people following the same recipe with the same ingredients routinely produce different results, not because the information differs, but because the organization of perception, motor control, and experience differs. When a cook leaves a kitchen, the knowledge of how that kitchen functions leaves with them, even if the recipe book remains behind.

Carpentry exhibits the same structure. A blueprint specifies dimensions and relations, but it does not encode the tacit knowledge required to select materials, account for warping, compensate for tool wear, or adjust to irregularities in a building site. The knowledge that allows a carpenter to frame a wall that is square, load-bearing, and durable is distributed across their body, their tools, and their familiarity with specific materials. When a crew dissolves, the capacity to build in a particular style or to a particular standard can vanish, despite the continued existence of drawings and codes.

Electrical and plumbing work further illustrate the irreducibly embodied and situated nature of knowledge. Wiring diagrams and plumbing schematics describe idealized systems, but real installations confront legacy infrastructure, idiosyncratic layouts, and accumulated modifications. Skilled electricians and plumbers rely on pattern recognition, sensory cues, and experiential heuristics to diagnose faults and prevent failures that cannot be anticipated in advance. The knowledge that keeps a building functional is not stored in documents but maintained through the ongoing presence of practitioners capable of responding to unforeseen conditions.

Childcare offers an even starker example. Manuals on child development do not contain the knowledge required to care for a particular child. That knowledge emerges from sustained interaction, attunement, and adaptation to a living system that changes from day to day. It is relational rather than transferable. When caregivers change, the knowledge of how to soothe, teach, or protect a child must be rebuilt through new patterns of interaction. Nothing meaningful has been “copied,” even though extensive information may be available.

Across these domains, a common structure appears. Knowledge is not an object that can be handed off intact, nor a set of propositions that can be stored indefinitely. It is a property of coordinated activity unfolding in time, dependent on bodies, tools, environments, and social roles. Instructions and records function only insofar as they support the reconstruction of this coordination, and they fail whenever such reconstruction exceeds local capacity.

These examples reveal why knowledge loss is so common and so poorly understood. When the organization that sustains skilled practice dissolves, the knowledge does not merely become harder to access; it ceases to exist in any actionable sense. This is not a defect of documentation but a consequence of embodiment. The same structure that governs cooking, carpentry, electrical work, plumbing, and childcare scales upward to firms, cities, and economies. The loss of collective capability follows the same logic as the loss of a craft tradition or a caregiving relationship. Knowledge disappears when the systems that enact it can no longer be held together.

### 3 The Person-Byte and Collective Cognition

If ordinary skilled practices reveal knowledge to be embodied, situated, and fragile, then the scale of modern production makes this condition unavoidable. César Hidalgo captures this inevitability with the notion of the *person-byte*, the finite upper bound on the amount of actionable knowledge any individual can embody. This bound is not merely cognitive, but physical, temporal, and social. Human beings have limited lifespans, limited attention, and limited capacity to maintain active coordination across many domains simultaneously. No individual, regardless of training or intelligence, can hold the full set of capabilities required to design, manufacture, and maintain complex modern artifacts such as aircraft, data centers, or pharmaceutical supply chains.

The consequence of the person-byte is that modern knowledge cannot reside within individuals. Any artifact whose production exceeds the capacity of a single person must distribute its required capabilities across a network. An aircraft, a semiconductor fabrication plant, or a national power grid is not the product of exceptional minds operating in isolation, but of architectures that decompose complex tasks into interlocking roles whose combined activity exceeds individual limits while

remaining functionally coherent. Knowledge therefore becomes irreducibly collective. It exists only insofar as a network of agents, each holding partial and specialized capabilities, can coordinate their actions in a way that preserves overall coherence.

From this perspective, firms, cities, and industries are not best understood as aggregations of workers or repositories of skills. They are computational architectures designed to overcome the person-byte constraint by partitioning capability across roles while maintaining functional integration. Job descriptions, professional licensing, organizational hierarchies, and technical standards are not secondary administrative layers. They are the computational substrate that allows bounded agents to function as a single productive unit.

This perspective clarifies why knowledge growth at the collective level differs qualitatively from individual learning. An individual can learn faster by studying harder or practicing longer, but a collective can only grow its capabilities by reorganizing itself. As systems scale, the dominant constraint shifts from individual acquisition to coordination overhead. Each additional person-byte introduced into a system increases not only capacity but also the burden of integration. Without architectures that stabilize interaction, added knowledge fragments rather than accumulates. Knowledge growth at the collective level is therefore inseparable from the design of coordination structures capable of preventing fragmentation.

The person-byte also explains why knowledge is lost so easily. When coordination breaks down, the distributed pieces of capability no longer compose into a functioning whole. The loss of a few key roles, tools, or interfaces can render an entire capability inoperable, even if most of the individual knowledge remains. When the networks that distribute and integrate capability are disrupted, the knowledge they sustain does not simply degrade gracefully. It collapses.

The dissolution of a firm, the emigration of key specialists, or the obsolescence of critical infrastructure can render previously routine capabilities impossible to reproduce, even when documentation remains intact. From the outside, this appears mysterious: how can a society forget how to build something it once produced routinely? From the perspective of collective cognition, the answer is straightforward. The knowledge did not reside in any single place. It existed only in the pattern of coordination that has since dissolved.

Understanding the person-byte reframes the problem of development and decline. Progress does not consist in filling minds with information, but in constructing and maintaining architectures that allow many limited agents to act as one. Development is not primarily the education of individuals in isolation, but the construction of systems that can coordinate many limited agents into a coherent whole.

The central question is not how much each person knows, but how effectively their partial knowledge can be composed into stable, reproducible capability. Likewise, decline is not primarily a matter of ignorance, but of organizational decay. When the structures that bind person-bytes together erode, knowledge evaporates, regardless of how much information remains archived.

## 4 The Scale-Dependence of Knowledge Growth

The constraints imposed by embodiment and collective coordination imply that knowledge growth cannot be scale-invariant. Hidalgo's analysis makes explicit that the dynamics governing learning at the level of individuals and small teams differ fundamentally from those governing growth at the level of industries and entire economies. Treating knowledge accumulation as a single homogeneous process obscures these differences and leads to persistent analytical and policy errors.

At the level of individuals and small groups, learning typically follows a power-law trajectory. Early progress is rapid because basic competencies yield large marginal gains. Over time, however, improvement slows as the learner approaches the limits imposed by the person-byte and by diminishing returns to practice within a fixed role. This plateau is not a failure of effort but a structural consequence of bounded embodiment. No amount of refinement can allow a single individual to absorb the full complexity of a task whose execution demands distributed specialization.

Industrial-scale growth operates in a qualitatively different regime. When knowledge is successfully distributed across stable architectures of coordination, collective capability can grow exponentially. This is the regime captured by Moore's Law and related industrial scaling phenomena. Crucially, this growth does not arise from faster individual learning, but from organizational innovation that preserves coherence as complexity increases. New interfaces, standards, and modular decompositions allow additional person-bytes to be added without overwhelming the system's capacity to integrate them.

This distinction explains why attempts to scale individual learning strategies often fail when applied to institutions or economies. Training programs, educational reforms, and skill subsidies can improve local competencies, but they cannot by themselves generate sustained growth unless accompanied by changes in organizational structure. Without architectures that absorb and coordinate new capabilities, accumulated knowledge saturates and fragments rather than compounding.

The scale-dependence of growth also clarifies why economic development is so uneven. Regions that succeed in creating coordination structures capable of supporting complex production enter a regime of accelerating returns, while those that fail remain trapped in slow, saturating growth. The difference is not primarily cultural or motivational, but architectural. Growth at scale is a property of systems that can continuously reorganize themselves to manage rising complexity without collapsing.

Understanding knowledge growth as scale-dependent reinforces the central thesis that development is an organizational achievement rather than a purely cognitive one. Sustained progress depends on the capacity to redesign coordination structures in response to increasing demands, not merely on the accumulation of skills or information. Where such redesign succeeds, exponential growth becomes possible. Where it fails, even substantial investments in education and training yield only modest and temporary gains.

## 5 Diffusion as Local Reconstruction

Knowledge is often described as diffusing across space, as though it were a substance capable of flowing freely from one location to another. This metaphor is deeply misleading. Knowledge does

not move in the manner of a fluid or a signal. It must be reconstructed within each new context in which it appears. The difficulty of diffusion arises precisely because knowledge is embodied and relational rather than symbolic.

Hidalgo's historical example of Samuel Slater illustrates this point with clarity. The transfer of textile manufacturing capability from Britain to the United States did not occur through the shipment of blueprints or the copying of manuals. It occurred through the physical relocation of a person who embodied the relevant knowledge. Even then, the knowledge did not arrive intact. It had to be adapted to new materials, new labor conditions, and new institutional constraints. Diffusion was not transmission but re-instantiation.

This pattern generalizes across domains. The movement of skilled workers, engineers, and practitioners is a primary driver of knowledge diffusion because they carry with them not just information, but the capacity to reconstruct coordinated activity in new environments. Written records and formal descriptions can support this process, but they cannot substitute for it. Where embodied experience is absent, diffusion stalls regardless of how much information is available.

Viewing diffusion as local reconstruction also explains why it is so path-dependent. New capabilities emerge most readily where existing structures already support related forms of coordination. Attempting to implant a complex industry into a region lacking the necessary complementary capabilities often results in failure, not because the knowledge is inaccessible, but because the local environment cannot sustain its reconstruction. Diffusion succeeds only when the receiving context can absorb and reorganize itself around the incoming capability.

This perspective challenges policy narratives that treat knowledge as a mobile input that can be injected at will. Investments in education, infrastructure, or technology transfer cannot produce complex capabilities unless they align with the existing organizational fabric of a region. Diffusion is slow, costly, and fragile because it requires the synchronized evolution of people, tools, and institutions.

Understanding diffusion as reconstruction rather than movement reinforces the broader thesis that knowledge is a property of organized systems. It does not travel independently of the structures that sustain it. Where those structures cannot be reproduced, diffusion fails. Where they can be carefully rebuilt, knowledge takes root and grows.

## 6 Relatedness as Structural Compatibility

If diffusion is a process of local reconstruction, then the ease or difficulty of that reconstruction depends on the degree of structural compatibility between existing capabilities and the new activity being attempted. Hidalgo formalizes this observation through the Principle of Relatedness, which holds that new forms of production emerge most readily from activities that share complementary inputs, skills, and coordination patterns with those already present.

Relatedness is not a superficial similarity between outputs, nor is it reducible to industrial classification schemes. It reflects deep overlaps in the organizational architecture required to sustain a capability. Two activities are related when the physical tools, social roles, regulatory structures, and tacit competencies required for one can be repurposed for the other with limited reconfiguration.

Where such overlap exists, reconstruction is feasible. Where it does not, the cost of coordination rises sharply.

This explains why economic diversification follows narrow and predictable paths. Regions do not leap arbitrarily into complex new industries. They expand into adjacent activities that reuse existing structures of coordination. A textile region may evolve into garment manufacturing or technical fabrics, but it cannot plausibly jump directly into semiconductor fabrication without first accumulating a vast array of intermediate capabilities. The limitation is not a lack of ambition or foresight, but the absence of compatible organizational substrates.

Relatedness also clarifies why attempts at rapid industrial transformation so often fail. Policies that ignore structural compatibility treat knowledge as a set of interchangeable inputs rather than as a tightly coupled system. When new industries are imposed without regard to local architectures, the resulting systems lack the internal coherence required to persist. They may function temporarily under external support, but they collapse once that support is withdrawn.

At a finer scale, relatedness governs innovation within firms and technologies. Incremental improvements succeed when they operate within an established architecture. Radical departures succeed only when accompanied by a redesign of the coordinating structures that hold the system together. Relatedness thus acts as a constraint on imagination itself, shaping which futures are reachable from a given present.

By framing relatedness as structural compatibility rather than similarity, the theory of knowledge diffusion acquires explanatory depth. It becomes possible to understand not only where growth is likely to occur, but why certain trajectories remain inaccessible despite apparent opportunity. Knowledge expands along paths that preserve coherence, and it resists jumps that would require coordination structures to change faster than they can be rebuilt.

## 7 Non-Fungibility and Architectural Innovation

The principle of relatedness ultimately rests on a more fundamental property of knowledge: its non-fungibility. Knowledge does not behave like a homogeneous quantity whose units can be substituted without consequence. The components of knowledge are specific, differentiated, and irreducibly contextual. Hidalgo's metaphor of an infinite alphabet captures this condition precisely. Each capability corresponds to a distinct letter, and no amount of one letter can substitute for the absence of another.

Non-fungibility explains why the accumulation of experience or expertise in one domain cannot compensate for gaps in another. A surplus of software engineers does not offset a shortage of chemical engineers, nor does excellence in marketing substitute for deficiencies in manufacturing. Knowledge components interlock, but they do not collapse into a single scalar measure. Productive capacity depends on possessing the right combinations, not merely large amounts.

This property becomes most visible in the contrast between incremental and architectural innovation. Incremental innovation modifies individual components within an existing organizational syntax. Because the underlying coordination structures remain intact, such changes can often be absorbed with minimal disruption. Architectural innovation, by contrast, redefines the syntax it-

self. It alters how components are connected, sequenced, and governed. In doing so, it changes the meaning of the existing components and renders prior forms of coordination obsolete.

Hidalgo's comparison between Amazon and Barnes & Noble illustrates this distinction vividly. Barnes & Noble refined the traditional retail architecture by optimizing inventory management and store layout. Amazon, by contrast, redesigned the entire architecture of book distribution, replacing storefronts with fulfillment centers and reconfiguring the flow of goods, information, and labor. The resulting distance between the two systems was not a matter of efficiency but of structure. Once the architecture changed, the capabilities embedded in the old system could not be smoothly adapted to the new one.

Non-fungibility thus imposes sharp limits on adaptation. When architectures change, knowledge tied to the old structure loses its operational meaning. This is not because the knowledge becomes false, but because it no longer fits within the new system of coordination. Architectural innovation therefore creates both extraordinary gains and irreversible losses. It expands the space of what is possible while simultaneously collapsing the value of capabilities that cannot be reembedded.

Understanding innovation through the lens of non-fungibility clarifies why technological revolutions are so disruptive. They do not merely introduce new tools; they reorder the alphabet itself. Those who control the new syntax gain outsized advantage, while those whose knowledge is bound to the old syntax face rapid obsolescence. Innovation is therefore not a smooth ascent, but a series of structural discontinuities that reshape the landscape of collective capability.

## 8 Measuring the Shadow of Capability

Because knowledge is embodied, distributed, and non-fungible, it resists direct measurement. There is no natural unit in which knowledge can be counted, no instrument that can register it independently of the systems that enact it. Attempts to quantify knowledge as years of education, numbers of patents, or volumes of information invariably fail to capture its operative structure. What can be measured, however, is the shadow that knowledge casts in the space of observable outcomes.

Hidalgo's contribution lies in recognizing that productive capabilities reveal themselves indirectly through patterns of specialization. By examining which products a country, region, or firm is able to produce competitively, one can infer the diversity and sophistication of the underlying capabilities that must be present. These observations do not expose knowledge itself, but they constrain the set of possible architectures consistent with observed performance.

The resulting measures of economic complexity are therefore not ontological claims about what a system knows. They are diagnostic tools that estimate how many distinct and complementary capabilities must be coordinated to generate the observed outputs. A region that produces a wide array of complex products must possess a large and well-integrated alphabet of capabilities, regardless of how those capabilities are distributed across individuals and institutions.

This approach also explains why complexity measures have predictive power. When a system exhibits a level of productive complexity that exceeds what would be expected given its current income or scale, it indicates latent capacity. Because the coordination structures required for complex production are already in place, the system is positioned to expand along related dimensions.

Growth in such cases does not require the invention of entirely new architectures, but the extension and recombination of existing ones.

At the same time, the indirect nature of these measures imposes limits on their interpretation. Complexity metrics cannot distinguish between fragile and robust forms of coordination, nor can they reveal how close a system is to structural breakdown. Two regions may exhibit similar levels of complexity while differing dramatically in their ability to maintain coherence under stress. Measurement captures the presence of capability, not its durability.

The notion of measuring the shadow of capability thus reinforces the central theme of this essay. Knowledge cannot be isolated, stored, or summed. It can only be inferred from sustained patterns of coordinated action. Metrics that respect this constraint illuminate the structural conditions of growth without succumbing to the illusion that knowledge itself has been captured.

## 9 Knowledge Loss as the Default State

The preceding analysis of embodiment, coordination, and measurement leads to a conclusion that is both counterintuitive and unavoidable: knowledge loss is not an exception to the normal course of economic and social life, but its default condition. Because knowledge exists only as a property of organized systems, it persists only so long as those systems remain intact. When coordination decays, knowledge does not merely become harder to access; it ceases to exist in any operational sense.

This fragility is often obscured by the persistence of archival artifacts. Blueprints, manuals, databases, and repositories give the appearance of durable knowledge storage. Yet these artifacts function only as supports for reconstruction. They do not themselves enact the coordinated activity that constitutes knowledge. When the social and technical networks required to interpret and apply these records dissolve, the records become inert. The capacity they once supported cannot be revived without rebuilding the missing organizational structures.

Historical examples abound. Entire classes of technologies, from ancient metallurgical techniques to early photographic processes, have been lost despite extensive documentation. The disappearance of these capabilities is not due to a failure of memory, but to the erosion of the collective arrangements that once sustained them. When apprenticeship chains break, when specialized tools are no longer manufactured, or when complementary industries vanish, the knowledge tied to those systems evaporates.

At the level of firms and industries, the same dynamic applies. Organizational restructuring, outsourcing, and short-term optimization can undermine the coordination patterns that support complex capabilities. When experienced teams are dispersed or when interfaces are redesigned without regard to tacit dependencies, firms may find themselves unable to reproduce outcomes that were routine only years earlier. Knowledge loss in such cases is often misdiagnosed as a skills gap or a training failure, when it is in fact a structural collapse.

Recognizing knowledge loss as the default state reframes the problem of development. Growth is not a matter of accumulating ever more knowledge atop a stable foundation. It is a continuous struggle against entropy, requiring constant investment in the maintenance of coordination. With-

out such investment, complex systems revert toward simpler forms, shedding capabilities as the cost of integration exceeds available resources.

This insight also clarifies why periods of rapid advancement are historically rare and fragile. They require sustained alignment between educational systems, organizational architectures, institutional norms, and material infrastructure. When that alignment falters, regression follows not as a catastrophe but as a natural consequence. Knowledge decays because the systems that held it together can no longer be maintained.

## 10 Knowledge as Maintained Coherence

If knowledge loss is the default condition, then knowledge growth must be understood not as accumulation but as the sustained maintenance of coherence under increasing complexity. From this perspective, knowledge is not something a system possesses once and for all. It is something a system continually reproduces through coordinated action. The central challenge of development is therefore not invention alone, but stabilization.

Maintained coherence refers to the capacity of a system to preserve functional integration among its components as the demands placed upon it intensify. Each new capability introduced into a productive system increases the burden on coordination. Additional roles must be synchronized, interfaces must be stabilized, and failure modes must be anticipated. Without compensating organizational adaptation, complexity overwhelms coherence and the system sheds capability.

This reframing resolves a long-standing ambiguity in discussions of productivity and innovation. High-performing systems are often described as more knowledgeable, but what distinguishes them is not the sheer quantity of information they contain. It is their ability to keep many differentiated activities aligned over time. Coherence is the scarce resource, not knowledge in the abstract. Where coherence is maintained, capabilities persist and compound. Where it is not, they fragment and disappear.

Maintained coherence also clarifies the role of institutions. Educational systems, professional standards, regulatory frameworks, and cultural norms function as stabilizing forces that reduce the entropy of coordination. They do not directly produce knowledge, but they lower the cost of maintaining it by providing shared expectations and reliable interfaces. When such institutions erode, the burden of coordination shifts onto individuals and organizations, often exceeding their capacity to cope.

This view casts innovation in a new light. Successful innovation is not merely the introduction of novelty, but the integration of novelty into an existing coherent system. Many technically viable ideas fail not because they lack merit, but because they impose coordination costs that exceed the system's ability to absorb them. Conversely, modest innovations can have transformative effects when they simplify coordination or reinforce existing structures of coherence.

Understanding knowledge as maintained coherence thus unifies growth, diffusion, and loss within a single framework. Growth occurs when coherence can be extended to encompass greater complexity. Diffusion succeeds when coherence can be reconstructed locally. Loss follows when coherence cannot be sustained. In all cases, the decisive variable is not information, but the capacity of

organized systems to hold together under strain.

## 11 Policy as Infrastructure Stewardship

If knowledge is a property of organized systems and development is the maintenance of coherence under complexity, then the role of policy must be reconceived accordingly. Traditional policy frameworks treat knowledge as an input to be supplied, accumulated, or incentivized. Education, research funding, and technology transfer are pursued as though they were deposits into a warehouse of capability. Such approaches consistently underperform because they misidentify the locus of constraint.

From the perspective developed here, policy does not create knowledge directly. It shapes the conditions under which collective coordination can be sustained. Its primary function is not accumulation but stewardship. Effective policy maintains and gradually extends the infrastructures that allow distributed capabilities to remain coherent over time.

This reframing explains why development initiatives so often fail despite substantial investment. When policies attempt to force rapid diversification into unrelated industries, they impose coordination demands that exceed local capacity. When they emphasize formal instruction without corresponding opportunities for embodied practice, they produce credentials without capability. When they prioritize short-term efficiency over institutional continuity, they erode the very structures that sustain complex production.

Infrastructure stewardship requires patience because coherence evolves slowly. Educational systems, professional pipelines, supply chains, and regulatory regimes must be aligned not only with present needs but with plausible future trajectories of related growth. The principle of relatedness thus becomes a guiding constraint on policy design. Successful interventions expand the space of feasible activity by reinforcing adjacent capabilities rather than attempting to leap across structural gaps.

This perspective also clarifies the political difficulty of development. The benefits of coherence maintenance are diffuse and long-term, while the costs are immediate and visible. Institutions that stabilize coordination often appear inefficient precisely because they absorb entropy that would otherwise manifest as failure. When such institutions are dismantled in the name of flexibility or innovation, the resulting loss of capability is typically recognized only after it has become irreversible.

Policy as infrastructure stewardship therefore demands a shift in evaluative criteria. Success should be measured not by the number of programs launched or the volume of resources allocated, but by the durability of collective capabilities under stress. Where coordination persists and adapts, knowledge grows. Where it fragments, no amount of investment can compensate.

## 12 Computation After Storage: A Structural Synthesis

The preceding sections converge on a structural redefinition of computation itself. Once knowledge is understood as an embodied, distributed, and fragile property of organized systems, the classi-

cal metaphor of computation as execution over stored state becomes inadequate. Storage-centric models presuppose that meaning can be frozen, copied, and restored without loss. The analysis developed here shows that this presupposition fails whenever computation involves semantic coordination across time, people, and institutions.

Computation after storage is not computation without records, but computation without the illusion that records are primary. In this regime, the fundamental objects of computation are not files or snapshots but histories of constrained transformation. State is not an ontological primitive but a projection, extracted from ongoing processes for the sake of local action. What appears as persistent state is in fact a stabilized pattern of activity continuously reproduced against entropy.

This reframing aligns computation with the dynamics observed in economic development and collective knowledge. Productive systems compute by maintaining coherence across distributed agents, tools, and norms. They process information not by symbol manipulation alone, but by regulating which transformations remain admissible as contexts evolve. Computation becomes inseparable from maintenance, repair, and adaptation.

The failure of storage-centric metaphors is most evident in systems that require merge. When divergent histories must be reconciled, no stored representation can determine the outcome without reference to external constraints and judgment. Merge is not an operation that can be applied mechanically to artifacts; it is a semantic event that resolves incompatibilities at the cost of irreversibility. Computation after storage therefore accepts loss, ambiguity, and partial coherence as structural facts rather than exceptional conditions.

This synthesis also clarifies why modern computational infrastructure increasingly resembles governance rather than machinery. Version control systems, protocols, and standards function less as repositories of truth than as arenas in which coordination is negotiated and stabilized. The success of such systems depends not on their ability to preserve information indefinitely, but on their capacity to sustain workable coherence under continual change.

Computation after storage thus provides a unifying lens through which the laws of knowledge growth, diffusion, and valuation can be understood. All three concern the same underlying problem: how organized systems maintain semantic coherence over time in the face of bounded embodiment and rising complexity. Storage was a useful abstraction under conditions that no longer obtain. In its place emerges a view of computation as irreversible semantic evolution, governed by constraints that cannot be wished away.

## 13 Computational Pathologies as Semantic Failure

The architectural distinction between event-historical and storage-centric computation is not merely theoretical. It predicts concrete performance differences in systems that appear functionally equivalent. Consider the case of gesture-based text input.

### The Swype Keyboard as Architectural Loss

A particularly revealing computational example of semantic loss through architectural regression is the case of the original Swype keyboard. Swype was among the earliest gesture-based text input

systems deployed at scale on mobile devices, and in several dimensions it remains superior to many contemporary trace or glide keyboards despite having been discontinued and partially imitated.

What distinguished Swype was not merely its ability to recognize continuous finger trajectories, but the semantic architecture underlying its recognition model. Swype treated a gesture as a single coherent semantic event whose meaning was inferred holistically from the entire trajectory, rather than as a sequence of locally decoded micro-events. The system implicitly modeled words as topological paths through a constrained semantic space defined by keyboard geometry, lexical priors, and user-specific adaptation.

Modern trace keyboards, by contrast, often reintroduce a storage-centric abstraction. They decompose a gesture into time-indexed samples, classify intermediate points, and then attempt to reconstruct intent through probabilistic aggregation. This shift appears innocuous, and it aligns well with contemporary machine learning pipelines, but it subtly abandons the event-historical semantics that made Swype robust.

The consequences are visible to experienced users. Swype exhibited lower error rates for long or uncommon words, greater stability under variation in speed and pressure, and more predictable behavior under partial occlusion or atypical trajectories. Crucially, it also demonstrated a form of semantic inertia: once a user established a stable gestural pattern for a word, the system resisted overcorrection, preserving learned intent even when local evidence was noisy.

In Spherepop terms, Swype’s gesture recognition operated over event scopes rather than decomposed traces. Each gesture constituted a single semantic commitment whose resolution depended on the entire trajectory history, not on local sampling. The system’s resistance to overcorrection reflects the irreversibility principle: once a gestural pattern had been committed to a particular semantic interpretation, that commitment could not be undone without discarding accumulated contextual information. Modern trace keyboards, by contrast, treat each time-step as independently revisable, which appears flexible but actually fragments semantic coherence across the gesture.

Under a storage-centric interpretation, these properties are difficult to explain. They appear as implementation details or tuning artifacts. Under a semantic-maintenance framework, however, they follow directly. Swype maintained a locally coherent mapping between gesture histories and semantic commitments, and it expended computational effort to preserve that coherence rather than continuously re-optimizing against a global statistical objective.

The discontinuation of Swype is therefore not merely a story of market forces or corporate acquisition. It is an instance of knowledge loss through architectural replacement. The semantic capability embodied in Swype was not fully captured by its codebase, patents, or published descriptions. When the coordination between its designers, its internal representations, and its maintenance practices dissolved, the capability could not be perfectly reconstructed, even though superficially similar systems proliferated.

This example illustrates a central claim of the paper: semantic performance is not monotonically improved by more data, larger models, or finer-grained traces. In some cases, those moves actively degrade performance by violating the constraints of semantic locality. What is lost is not information in the narrow sense, but a stable architecture for interpreting history as a meaningful whole.

The Swype case is representative of a broader class of regressions that occur when event-historical architectures are replaced by storage-centric approximations that appear more tractable but violate semantic locality constraints.

### **Autocorrect Drift as Coordination Collapse**

A related pathology appears in modern autocorrect systems, which increasingly exhibit temporal instability despite ostensibly learning from user behavior. Users report that corrections which were reliably suppressed after explicit rejection begin reappearing after system updates, that personalized vocabulary degrades unpredictably, and that the boundary between helpful correction and aggressive override shifts without user action.

This instability follows directly from treating correction as a stateless optimization problem. Each correction decision is made by maximizing local probability against a continuously updated global model, with user corrections treated as training signals rather than binding commitments. The system has no architectural representation of the constraint that a user has rejected a given correction in a given context and that this rejection must be preserved.

In event-historical terms, autocorrect systems lack constraint persistence. User corrections are events that should establish durable semantic commitments, namely constraints that narrow the space of admissible future corrections. Instead, these events are converted into statistical adjustments that can be overridden by subsequent model updates, A/B tests, or shifts in the training distribution.

The result is that users must repeatedly re-teach the system the same distinctions, because the coordination between user intent and system behavior cannot be maintained across architectural boundaries. What appears as a learning system is in fact failing to preserve learned constraints, exhibiting the fragility that the semantic maintenance framework predicts when event history is discarded in favor of reconstructed state.

## **14 Historical Reconstruction and Knowledge Loss**

The claim that knowledge loss is the default state and that knowledge exists only as a property of maintained coordination is not an abstract philosophical position. It is a structural prediction that should be visible across historical contexts whenever the architectures sustaining complex capabilities dissolve. Two cases separated by nearly two millennia demonstrate this pattern with particular clarity: the loss of Roman concrete formulation and the erosion of Boeing 747 manufacturing and modification capability.

### **Roman Concrete as Coordination Collapse**

Roman concrete, particularly the opus caementicium used in major imperial structures, exhibited durability properties that were not reliably reproduced for over a millennium after the collapse of the Western Roman Empire. Modern analysis reveals that these properties derived from specific volcanic pozzolana sources, precise water-to-lime ratios, careful aggregate selection, and curing

protocols adapted to maritime or terrestrial contexts. Extensive written records survived, including Vitruvius's *De Architectura*, which describes mixing procedures, material selection, and structural applications in considerable detail.

Yet despite the persistence of these texts, the capability to produce Roman concrete at scale disappeared. This is often mischaracterized as a mystery of lost recipes or forgotten ingredients. The semantic maintenance framework reveals a different structure. Roman concrete was not a recipe that could be executed by anyone with access to instructions. It was a capability sustained by a network of coordinated practices: quarrying operations that could identify suitable pozzolana deposits by texture and color, transport systems that could move materials economically over distances, construction guilds that could judge mixture consistency by hand, institutional demand structures that made large-scale concrete construction economically viable, and apprenticeship chains that transmitted tacit adjustments across generations.

In the terms established earlier in this essay, Roman concrete corresponds to a semantic locality whose coherence depended on the simultaneous satisfaction of architectural, material, economic, and organizational constraints. Vitruvius's text captures only a projection of this locality onto the space of symbolic instructions. The projection is accurate within its scope, but it cannot reconstruct the full capability because it does not encode the embodied judgments, the supply chain coordination, or the institutional context that made those instructions actionable.

When the Western Empire fragmented, the coordination structure dissolved. Quarries were abandoned or repurposed, long-distance material flows became uneconomical, construction guilds lost continuity, and institutional patrons who understood what Roman concrete could accomplish disappeared. The capability did not fade because knowledge was forgotten in the sense of information loss. It collapsed because the network of constraints that held the distributed capability together could no longer be maintained. What remained were texts that could not be executed and ruins that could not be replicated.

This case confirms a key prediction of the framework: when a capability exceeds the person-byte and depends on coordination across agents, tools, and institutions, its loss follows directly from coordination collapse. The persistence of documentation is irrelevant if the structures required to interpret and enact that documentation have dissolved. Roman concrete was lost not once but continuously, as each generation inherited texts without inheriting the systems that could make those texts meaningful.

## Boeing 747 as Layered Capability Erosion

A modern industrial example provides a more finely grained view of the same phenomenon. The Boeing 747, introduced in 1969, was manufactured continuously in various configurations for over five decades. Production of new aircraft ceased in 2022. During the program's active years, Boeing possessed not only the ability to manufacture the 747 but also the ability to make certain classes of design modifications that were routine in the 1970s through the 1990s but became progressively more difficult in later decades.

The most visible layer of capability loss concerns tooling and manufacturing knowledge. The 747's fuselage sections, wing assemblies, and structural joints were produced using specialized tool-

ing, much of it custom-designed for the program. Experienced workers developed tacit knowledge about how to compensate for tool wear, how to interpret dimensional variation, and how to sequence operations to avoid downstream defects. As production rates declined and the workforce aged, this knowledge was not systematically transferred, in part because it could not be fully formalized. Tribal knowledge about "shimming" techniques, fastener torque judgment under temperature variation, and visual inspection of complex joints resided in the coordinated practices of specific teams.

When those teams dissolved through attrition, retirement, or reassignment, the capability to manufacture the 747 to original quality standards degraded. This occurred despite the existence of extensive engineering drawings, work instructions, and quality control procedures. The documentation remained intact, but the organizational structures required to interpret ambiguities, resolve conflicts between specification and physical reality, and maintain consistency across production runs eroded. Attempts to restart production after extended pauses revealed that even with complete documentation, reconstructing manufacturing capability required costly and time-consuming retraining that could not fully recover the original performance.

A deeper layer of capability loss concerns design iteration. During the 1970s and 1980s, Boeing could introduce modifications to the 747 with relative speed and confidence. New variants, engine configurations, and structural reinforcements were designed, tested, and certified in a context where the design teams had living memory of earlier decisions, where interfaces between subsystems were governed by stable interpretive conventions, and where the consequences of changes could be evaluated by engineers who had participated in prior iterations.

As personnel turnover increased and as the institutional memory of the program aged, the ability to make modifications safely and efficiently declined. Changes that would have been straightforward in 1980 required extensive analysis and validation in 2010, not because the technical challenges were greater, but because the coordination structures that once held design knowledge together had degraded. Engineers working on later variants did not have direct access to the reasoning behind earlier design choices, and documentation, while extensive, did not capture the constraints and trade-offs that shaped those choices. The capability to modify the 747 was thus not merely a matter of technical skill but of coordination across design history.

In the semantic maintenance framework, the 747 program corresponds to a high-dimensional semantic locality maintained by stable architectural boundaries, personnel continuity, and institutional practices. Manufacturing capability and design-iteration capability are not independent; both depend on the preservation of constraint structures that govern admissible transformations. When those structures degrade, the capability does not disappear suddenly but erodes progressively, exhibiting the signature of increasing entropy cost for actions that were once routine.

The Boeing case also illustrates the non-fungibility of knowledge. The skills required to manufacture the 747 cannot be straightforwardly transferred to other aircraft programs, because they are coupled to the specific geometry, materials, and tooling of the 747 itself. Similarly, design knowledge is bound to the architectural decisions that defined the program. Attempts to reuse 747 knowledge in other contexts encounter the same obstacles as attempts to reconstruct it: the knowledge is inseparable from the systems that enacted it.

## Structural Lessons Across Contexts

Both cases reveal the same underlying pattern. Knowledge loss is not primarily a failure of memory or documentation. It is a consequence of coordination collapse. When the networks that distribute and integrate capability across people, tools, and institutions dissolve, the capability ceases to exist operationally, regardless of what has been written down. Reconstruction is possible in principle but requires not the retrieval of information but the rebuilding of coordination structures, often at prohibitive cost.

These examples also confirm that knowledge loss is scale-dependent and irreversible in practice. Roman concrete required coordination at the scale of an empire; when that empire fragmented, the capability could not be sustained at smaller scales. The 747 required coordination at the scale of a large aerospace corporation with stable institutional memory; when that memory degraded, the capability could not be fully recovered even within the same organization.

Most importantly, both cases demonstrate that the semantic maintenance framework is not a speculative theory but a descriptive account of phenomena that have repeatedly occurred throughout history. The laws articulated by Hidalgo, the computational structures formalized in Spherepop, and the field-theoretic foundations provided by RSVP all converge on the same conclusion: knowledge persists only where coordination persists, and its loss is the default outcome when structures of alignment fail.

## 15 Implications for Intelligence and Agency

Recasting knowledge as maintained coherence within organized systems has direct implications for how intelligence and agency are to be understood. Intelligence is often treated as the possession of an internal model that accurately represents the world. Agency, in turn, is framed as the capacity to act upon that model to achieve desired outcomes. Both notions presuppose that knowledge can be localized within an individual or a system as a stable representational asset.

The framework developed here suggests a different interpretation. Intelligence is not the possession of a global model, but the capacity to navigate constraints locally while preserving coherence over time. An intelligent system is one that can remain viable as conditions change, selectively incorporating new information while avoiding transformations that would destabilize its internal coordination. What matters is not completeness of representation, but adequacy of projection.

Agency emerges from this same capacity. To act is to commit to a trajectory through a space of admissible transformations, knowing that such commitment is irreversible and costly. An agent does not choose from a menu of fully specified options; it constructs futures incrementally, guided by local constraints and limited foresight. The exercise of agency therefore consists in managing entropy, not in optimizing over a static landscape.

This view dissolves the sharp boundary often drawn between cognition and infrastructure. Human intelligence relies on external supports such as language, institutions, tools, and shared practices that stabilize coordination and extend the effective reach of the person-byte. Likewise, organizational and societal forms of agency arise when these supports are arranged in ways that allow collective commitments to be made and sustained. Intelligence scales not by accumulating

representations, but by embedding agents within architectures that reduce the cost of coherent action.

The fragility of agency follows directly from this interpretation. When the structures that support coordinated projection degrade, intelligent behavior becomes brittle or impossible. Individuals may retain their skills, but without the surrounding systems that give those skills context and consequence, agency collapses into improvisation or paralysis. Failures of agency are therefore often misdiagnosed as failures of will or understanding, when they are in fact failures of infrastructure.

Understanding intelligence and agency as properties of maintained coherence aligns them with the broader theory of knowledge articulated in this essay. They are not exceptions to the laws governing collective capability, but expressions of them. To enhance intelligence or agency is not to enrich internal models, but to strengthen the architectures that allow bounded agents to act coherently under constraint.

## 16 Limits of Automation and the Role of Judgment

The preceding account of computation, knowledge, and agency places principled limits on what automation can achieve. If knowledge is embodied in coordinated systems and maintained through ongoing repair, then no algorithm can fully replace the judgment required to sustain coherence under changing conditions. Automation excels where constraints are stable, interfaces are well-defined, and the cost of error can be bounded in advance. It fails where coordination must be renegotiated, meanings must be reconciled, and losses must be chosen rather than avoided.

This limitation is not a temporary technological shortcoming but a structural fact. Automated systems operate by applying transformations whose admissibility has been specified in advance. When divergent histories collide or when contexts shift in unanticipated ways, admissibility itself becomes the question at issue. Resolving such situations requires commitments that cannot be derived from prior rules without circularity. Judgment intervenes precisely where rules run out.

Human judgment functions, in this framework, as an entropy-management mechanism. It selects which inconsistencies to tolerate, which constraints to relax, and which histories to privilege when coherence cannot be preserved in full. These choices are irreversible and value-laden. They cannot be automated without smuggling in an implicit authority that merely displaces the site of judgment rather than eliminating it.

The role of automation is therefore properly understood as local acceleration. By reducing the cost of routine coordination, automated systems free human judgment to operate where it is most needed. Problems arise when automation is mistaken for global resolution, and when systems are designed under the assumption that judgment can be eliminated rather than supported. In such cases, automation amplifies fragility by extending coordination structures beyond the contexts in which they can be maintained.

This analysis also clarifies recurring failures in large-scale technical systems. Crises attributed to human error often reflect the opposite condition: the withdrawal of human judgment from systems that had silently relied upon it. When automated infrastructures are pushed to operate beyond their designed constraints, breakdowns expose the irreducible need for interpretive intervention.

Limits of automation thus follow directly from the ontology of knowledge developed in this essay. Where knowledge is non-fungible, embodied, and irreversible, judgment is not a defect to be engineered away but a necessary component of any system that aspires to remain coherent over time.

## 17 Spherepop as Post-Storage Computation

The theoretical shift from storage-centric computation to irreversible semantic maintenance requires not only a new ontology but a corresponding computational form. Spherepop provides such a form by abandoning the assumption that computation operates over stable objects and replacing it with an event-historical calculus in which all values are already resolved contexts. In Spherepop, there are no bare primitives, no timeless states, and no operations that act independently of history. Every computation is a transformation of a bounded semantic sphere.

This structure directly addresses the failures identified in storage-based systems. Because Spherepop treats all values as scopes rather than atoms, it eliminates the distinction between data and context that underwrites the illusion of lossless copying. A Spherepop value cannot be duplicated without also duplicating its constraints, provenance, and authorization history. What appears as copying in conventional systems is revealed as reconstruction under constraint, with all the attendant costs and limits.

Merge in Spherepop is not defined as a binary operation over values but as a negotiation between spheres whose internal commitments may be partially incompatible. Because spheres are already resolved, merge cannot be commutative or reversible in general. It is instead an explicit act of commitment that produces a new sphere while discarding alternatives that can no longer be sustained. This makes the irreversibility of merge a first-class semantic fact rather than an error condition.

Spherepop therefore operationalizes the principle that computation after storage is computation over admissible histories. Logs are not auxiliary artifacts but constitutive of meaning. State is always derived, always provisional, and always relative to an observer’s position within a network of constraints. In this way, Spherepop implements at the level of programming language design the same insight that Hidalgo articulates at the level of economic development: capability exists only where coordination is actively maintained.

Most importantly, Spherepop provides a computational substrate in which knowledge loss is explicit rather than hidden. When a merge discards a possibility, that loss is represented in the resulting sphere’s structure rather than masked by overwriting or silent conflict resolution. Computation thus becomes an honest accounting of semantic cost, aligning software practice with the physical and economic realities described throughout this essay.

## 18 Derivation from Relativistic Scalar–Vector Plenum Theory

While the preceding analysis has proceeded at the level of economic structure, computation, and coordination, its core claims admit a deeper derivation from first principles. Relativistic Scalar–

Vector Plenum theory provides such a foundation by treating reality itself as a continuous field of scalar densities, vectorial flows, and entropy gradients rather than as a collection of discrete objects evolving over time.

Within RSVP, scalar fields represent locally accumulated structure, vector fields represent directed transformations and causal flow, and entropy quantifies the irreversible cost of reconfiguration. From this perspective, knowledge is not an epiphenomenon layered atop physical processes. It is a particular regime of field organization in which scalar coherence is locally stabilized against entropic dispersion by sustained vectorial alignment.

The person-byte constraint emerges naturally in this framework. Any bounded agent corresponds to a finite region of the plenum with limited capacity to stabilize scalar structure. Complex capabilities therefore require extended regions of coherence spanning multiple agents, tools, and institutions. Knowledge becomes collective not by convention, but by necessity, because no finite region can support the scalar density required for modern production on its own.

Diffusion, understood as local reconstruction, corresponds to the propagation of field coherence across regions that already possess compatible background structure. Where vector flows encounter incompatible scalar configurations, entropy rises sharply and reconstruction fails. This provides a physical analogue of the principle of relatedness, grounding it in field compatibility rather than empirical regularity alone.

Non-fungibility follows directly from the geometry of the plenum. Scalar structures cannot be arbitrarily substituted because they occupy distinct regions of configuration space. Architectural innovation corresponds to a topological reorganization of field alignment, producing new basins of coherence while rendering previous configurations unstable or inaccessible. Knowledge loss is thus not mysterious; it is the relaxation of unsupported scalar structure back into the plenum.

Seen through RSVP, computation after storage is not a technological trend but a return to physical honesty. Storage metaphors correspond to fictitious conserved quantities that ignore entropy and history. Spherepop’s event-historical semantics, like Hidalgo’s laws of knowledge, arise inevitably once computation is understood as constrained field evolution rather than symbolic manipulation.

This derivation completes the synthesis. Hidalgo identifies the empirical laws of knowledge in economies, Spherepop provides a computational form consistent with those laws, and RSVP supplies the ontological ground from which all three follow. What appears as economic development, software merge, or collective learning is, at base, the same phenomenon: the local maintenance of coherence in an entropic universe.

## 19 Structural Errors in Development Policy

If knowledge is a property of maintained coordination rather than an accumulable substance, then the persistent failure modes of development policy admit a structural explanation. These failures do not arise primarily from insufficient investment, poor incentives, or inadequate technical expertise. They arise from ontological misclassification. Policy frameworks that treat knowledge as a transferable input rather than as an embodied, distributed capability will systematically produce

outcomes that appear puzzling or disappointing but are in fact fully determined by the structure of the intervention.

From the perspective developed in this essay, development policy is not the management of inputs but the stewardship of semantic infrastructure. When policy acts as though it can inject knowledge into a system without sustaining the coordination structures that give that knowledge operational meaning, it creates conditions under which apparent progress masks long-term fragility.

One common structural error consists in attempting capability jumps that violate relatedness constraints. Policies often aim to move regions or organizations directly into high-complexity activities by importing advanced technologies, establishing specialized institutions, or mandating participation in globally competitive sectors. Such interventions assume that knowledge can be transplanted independently of the local coordination architectures that originally sustained it. The result is frequently the creation of isolated enclaves of technical activity that cannot integrate with the surrounding economy, require continuous external support, and fail to generate endogenous growth.

Within the semantic maintenance framework, these outcomes are not accidental. Capabilities emerge through adjacent reconstruction, not through leaps. When a new activity shares insufficient structural overlap with existing practices, the entropy cost of maintaining coherence exceeds what the local system can sustain. The intervention therefore produces either brittle success that collapses once support is withdrawn or nominal success that never acquires operational depth. What appears as stagnation is in fact a failure to respect the topology of semantic locality.

A second structural error lies in substituting credentials or symbolic indicators for coordination capacity. Educational expansion, certification programs, and formal training initiatives are often justified as mechanisms for increasing human capital. While such measures can play a role in capability formation, they frequently treat knowledge as something that can be instantiated at the level of individuals independently of the systems within which those individuals must operate. Degrees, certificates, and standardized curricula function as proxies for knowledge without guaranteeing the existence of environments where that knowledge can be enacted.

The person-byte constraint makes this error particularly costly. Complex capabilities require distributed cognition across teams, tools, and institutions. When policy focuses on producing credentialed individuals without simultaneously maintaining the coordination structures that allow those individuals to participate in collective problem-solving, it creates a surplus of latent capacity that cannot be activated. The resulting frustration is often misdiagnosed as a labor market mismatch or a failure of aspiration, when it is in fact a failure of semantic infrastructure.

A third error concerns the construction of infrastructure without maintenance pathways. Physical infrastructure, regulatory frameworks, and institutional arrangements are frequently treated as durable investments whose benefits accrue automatically over time. This assumption mirrors the storage-centric view of computation, in which state persists unless actively modified. In reality, infrastructure that supports complex coordination requires continuous semantic maintenance. Interfaces drift, practices evolve, and constraints change. Without ongoing alignment, infrastructure becomes miscalibrated to the capabilities it was intended to support.

This pattern is visible in industrial parks that fail to attract sustained activity, research institu-

tions that lose relevance despite continued funding, and regulatory regimes that become obstacles rather than enablers of innovation. The failure is not that infrastructure was built incorrectly, but that it was treated as a static artifact rather than as a living constraint system whose coherence must be actively preserved.

Across these cases, the unifying error is the same. Policy treats knowledge as a stock that can be accumulated, transferred, or stored, rather than as a process that must be continuously enacted under constraint. Once this misclassification is corrected, a different set of design principles becomes visible. Interventions must focus on maintaining coordination across existing practices, reducing the entropy cost of adjacent capability expansion, and preserving the interpretive continuity that allows distributed systems to function over time.

The implication is not that development policy is futile, but that its scope and nature have been misunderstood. Successful policy does not force systems toward desired end states. It stabilizes the semantic conditions under which coherent evolution becomes possible. Where those conditions are absent, no amount of investment or instruction can substitute for the slow work of rebuilding coordination structures.

In this sense, policy failure is not primarily a failure of execution. It is a failure to respect the physical and semantic laws governing how knowledge comes into being, how it persists, and how it is lost. Development strategies that ignore these constraints will continue to produce outcomes that appear paradoxical, while those that align with them will appear conservative even as they enable durable transformation.

## 20 Unified Substrates for Semantic Maintenance

The preceding sections have shown that failures in economic development, computational systems, and institutional policy share a common structure. In each case, capability depends on the maintenance of distributed coordination under irreversible constraint. When that coordination degrades, the capability erodes regardless of how much information, capital, or technical sophistication remains. What differs across domains is not the nature of the problem, but the substrate on which it is instantiated.

Spherepop and RSVP provide formal substrates in which this structure can be expressed directly. They do not introduce new empirical assumptions. Instead, they supply representations in which the dynamics already observed become natural consequences of the system's geometry.

Spherepop formalizes semantic activity as event-historical computation. Meaning is not stored state but arises from irreversible commitments made within bounded contexts. Transformations are admissible only insofar as they preserve coherence relative to prior commitments, and merge is necessarily costly because it requires reconciliation between incompatible histories. In this framework, semantic locality is explicit: what can be made coherent depends on the scope of the event context, and global consistency is neither assumed nor required.

This directly accounts for the computational pathologies examined earlier. Systems such as Swype succeed when they operate on coherent event scopes and fail when those scopes are fragmented into revisable traces. Autocorrect drift, version control conflicts, and interface instability

are not anomalies but expected outcomes when systems discard event history in favor of reconstructed state. Spherepop does not eliminate these costs; it makes them explicit and therefore tractable.

RSVP provides a complementary description at a more abstract level. Systems are modeled as scalar-vector-entropy fields in which structure persists only through constrained flow and continuous dissipation. Scalar quantities correspond to localized semantic density, vector fields correspond to admissible transformations, and entropy tracks the cost of maintaining coherence. Within this representation, irreversible loss is not a failure mode but the default trajectory absent sustained constraint enforcement.

This field-theoretic view aligns naturally with the empirical laws identified by Hidalgo. The person-byte limit corresponds to local scalar saturation. The principle of relatedness reflects the requirement that new flows align with existing field structure. Non-fungibility arises because capabilities correspond to specific field configurations that cannot be substituted without distortion. Knowledge loss occurs when the energetic cost of maintaining a configuration exceeds the system’s capacity to supply it.

Together, Spherepop and RSVP provide a unified description of semantic maintenance across scales. Economic development, computational semantics, and institutional policy differ in material realization but not in underlying dynamics. In each case, success depends on sustaining coherent structures against entropic pressure, and failure follows predictably when coordination pathways collapse.

Viewed through this lens, the persistence of capability is not a matter of accumulation but of geometry. Systems endure not because they store more information, but because they maintain the constraints that make coordinated action possible over time.

## 21 Conclusion

This essay has argued that the central problem of modern computation, economic development, and collective intelligence is not the accumulation of knowledge, but the maintenance of coherence under irreversible constraint. César Hidalgo’s laws, when taken seriously, do not merely describe empirical regularities of growth and diffusion. They expose a deeper ontology in which knowledge is a property of organized matter, sustained only through embodied coordination and continual repair.

Seen from this perspective, the historical reliance on storage-centric models of computation appears as a temporary scaffold rather than a foundation. Files, snapshots, and repositories offered workable abstractions under conditions of locality, stability, and low semantic friction. Those conditions no longer hold. Distributed systems, collaborative production, and rapid architectural change have made the limits of reversible abstraction unavoidable. Meaning cannot be stored without loss, merge cannot be automated without judgment, and diffusion cannot occur without reconstruction.

The synthesis developed here reframes these limitations as structural facts rather than engineering failures. Knowledge grows when systems succeed in extending coherence across greater

complexity. It diffuses when compatible structures allow local reconstruction. It is lost whenever the architectures that sustain coordination decay. These dynamics operate at every scale, from craft practices and caregiving to firms, cities, and global economies.

Spherepop demonstrates that these insights are not merely descriptive. They can be realized computationally. By treating all values as resolved contexts and all computation as event-historical transformation, Spherepop abandons the fiction of lossless storage and makes semantic cost explicit. In doing so, it aligns programming practice with the physical and economic realities of irreversible coordination.

Relativistic Scalar–Vector Plenum theory completes the picture by grounding this entire framework in first principles. When reality is understood as constrained field evolution rather than symbolic manipulation, the irreversibility of computation, the non-fungibility of knowledge, and the necessity of collective organization follow inevitably. What Hidalgo observes empirically, and what Spherepop enacts computationally, RSVP derives ontologically.

The resulting view is neither pessimistic nor utopian. It rejects the fantasy of perfect preservation and frictionless growth, but it affirms the possibility of sustained development through careful stewardship of structure. Progress becomes a matter of architectural patience rather than heroic accumulation. Intelligence becomes a property of maintained projection rather than exhaustive representation. Policy becomes an exercise in preserving adjacent coherence rather than forcing leaps across incompatible domains.

Computation after storage is therefore not a retreat from rigor, but an advance toward honesty. It accepts entropy, history, and loss as constitutive features of meaningful systems. In doing so, it offers a unified account of knowledge, computation, and development that treats infrastructure not as a toolset, but as the physics of collective capability itself.

## A Semantic Locality and Constraint Satisfaction

**Definition 1** (Constraint Space). *Let  $C$  be a set of semantic constraints. A constraint space is a pair  $(S, C)$  where  $S$  is a set of admissible semantic states.*

**Definition 2** (Satisfaction Relation). *A satisfaction relation is a relation*

$$\vdash \subseteq S \times C$$

*such that  $s \vdash c$  denotes that state  $s$  satisfies constraint  $c$ .*

**Definition 3** (Admissible Transformation). *A transformation  $t : S \rightarrow S$  is admissible if for all  $s \in S$  and for all  $c \in C$ , whenever  $s \vdash c$  and  $t(s)$  is defined, then  $t(s) \vdash c$ .*

**Definition 4** (Entropy Cost). *An entropy cost is a function*

$$\Delta : \mathcal{T} \times S \rightarrow \mathbb{R}_{\geq 0}$$

*assigning a nonnegative real value to each admissible transformation-state pair.*

**Definition 5** (Semantic Locality). *A semantic locality is a tuple*

$$\mathcal{L} = (S, \mathcal{T}, C, \vdash, \Delta)$$

*such that  $\mathcal{T}$  is closed under composition where defined and  $\Delta$  is bounded above by a finite constant on  $\mathcal{T} \times S$ .*

**Proposition 1** (Local Closure). *If  $s \in S$  and  $t_1, t_2 \in \mathcal{T}$  are admissible with  $t_2(t_1(s))$  defined, then  $t_2 \circ t_1 \in \mathcal{T}$ .*

**Theorem 1** (Constraint-Induced Admissibility). *For fixed  $(S, C, \vdash)$  and entropy bound  $\epsilon > 0$ , the set*

$$\mathcal{T}_\epsilon = \{t : S \rightharpoonup S \mid t \text{ admissible and } \Delta(t, s) < \epsilon\}$$

*is uniquely determined.*

**Proposition 2** (Non-Globality). *There exists no transformation  $t : S \rightarrow S$  admissible for all constraint extensions  $C' \supsetneq C$ .*

## B Entropy, Irreversibility, and Merge

**Definition 6** (Computational Entropy). *Let  $T : S \rightharpoonup S$  be a partial transformation. The computational entropy associated with  $T$  at state  $s$  is a quantity*

$$\Delta S(T, s) \in \mathbb{R}_{\geq 0}$$

*representing the minimal entropy production required to physically realize  $T(s)$  from  $s$ .*

**Definition 7** (Many-to-One Transformation). *A transformation  $T : S \rightarrow S$  is many-to-one if there exist distinct states  $s_1, s_2 \in S$  such that  $T(s_1) = T(s_2)$ .*

**Theorem 2** (Irreversibility of Constraint-Preserving Computation). *Let  $T : S \rightarrow S$  be a many-to-one transformation that preserves all constraints satisfied by  $s$  and is physically realizable with finite energy. Then*

$$\Delta S(T, s) > 0.$$

**Definition 8** (Merge Transformation). *A merge transformation is a partial function*

$$M : S \times S \rightharpoonup S$$

*such that for  $s_1, s_2 \in S$ , if  $M(s_1, s_2)$  is defined then it satisfies all constraints jointly satisfied by  $s_1$  and  $s_2$ .*

**Definition 9** (Lossless Merge). *A merge  $M$  is lossless if there exists an injective map*

$$\iota : S \times S \hookrightarrow S$$

*such that  $M = \pi \circ \iota$  for some projection  $\pi$ .*

**Theorem 3** (Impossibility of Perfect Merge). *There exists no merge transformation  $M : S \times S \rightarrow S$  that is simultaneously lossless, reversible, fully automated, and constraint-preserving for all admissible semantic states.*

**Proposition 3** (Entropy Increase Under Merge). *For any nontrivial merge  $M$  and admissible states  $s_1, s_2$ , the entropy production satisfies*

$$\Delta S(M, (s_1, s_2)) > 0.$$

## C Sheaf-Theoretic Semantics and Obstruction

**Definition 10** (Context Poset). *Let  $(X, \leq)$  be a partially ordered set whose elements represent semantic contexts and whose order relation represents contextual refinement.*

**Definition 11** (Semantic Presheaf). *A semantic presheaf  $\mathcal{F}$  on  $(X, \leq)$  assigns to each context  $U \in X$  a set  $\mathcal{F}(U)$  of semantic states and to each relation  $U \leq V$  a restriction map*

$$\rho_{V,U} : \mathcal{F}(V) \rightarrow \mathcal{F}(U)$$

such that  $\rho_{U,U} = \text{id}$  and  $\rho_{W,U} = \rho_{V,U} \circ \rho_{W,V}$  whenever  $U \leq V \leq W$ .

**Definition 12** (Compatible Sections). *Given contexts  $U, V \in X$  with  $W \leq U$  and  $W \leq V$ , sections  $s_U \in \mathcal{F}(U)$  and  $s_V \in \mathcal{F}(V)$  are compatible on  $W$  if*

$$\rho_{U,W}(s_U) = \rho_{V,W}(s_V).$$

**Definition 13** (Sheaf Condition). *A presheaf  $\mathcal{F}$  satisfies the sheaf condition if for every context  $V \in X$  and family  $\{U_i\}_{i \in I}$  with  $U_i \leq V$ , any family of sections  $\{s_i \in \mathcal{F}(U_i)\}$  that are pairwise compatible admits a unique section  $s \in \mathcal{F}(V)$  such that  $\rho_{V,U_i}(s) = s_i$  for all  $i$ .*

**Definition 14** (Global Section). *A global section of  $\mathcal{F}$  is an element of  $\mathcal{F}(X)$ , if such an element exists.*

**Theorem 4** (Obstruction to Global Coherence). *If there exist contexts  $U, V \in X$  and sections  $s_U \in \mathcal{F}(U)$ ,  $s_V \in \mathcal{F}(V)$  such that  $s_U$  and  $s_V$  are incompatible on some  $W \leq U, V$ , then no global section of  $\mathcal{F}$  exists.*

**Definition 15** (Čech Cohomology). *Let  $\{U_i\}$  be a family of contexts. Define the cochain groups*

$$C^0 = \prod_i \mathcal{F}(U_i), \quad C^1 = \prod_{i,j} \mathcal{F}(U_i \wedge U_j).$$

**Definition 16** (Čech Coboundary). *The coboundary operator  $\delta : C^0 \rightarrow C^1$  is defined by*

$$\delta(s)_{i,j} = \rho_{U_i, U_i \wedge U_j}(s_i) - \rho_{U_j, U_i \wedge U_j}(s_j).$$

**Theorem 5** (Cohomological Obstruction). *A family of sections  $\{s_i\}$  admits a global section if and only if*

$$\delta(s) = 0.$$

*Nonvanishing cohomology classes represent irreducible semantic conflict.*

## D Event-Historical Computation and State Projection

**Definition 17** (Event). *An event is an ordered pair*

$$e = (t, \tau)$$

*where  $t \in \mathbb{R}_{\geq 0}$  denotes occurrence time and  $\tau \in \mathcal{T}$  denotes an admissible transformation.*

**Definition 18** (History). *A history is a finite or countable sequence of events*

$$H = (e_1, e_2, \dots)$$

*such that the target state of  $e_i$  is the source state of  $e_{i+1}$ .*

**Definition 19** (Event-Historical System). *An event-historical system is a tuple*

$$\mathcal{E} = (S, \mathcal{T}, H)$$

*where  $S$  is a state space,  $\mathcal{T}$  is a set of admissible transformations, and  $H$  is a history.*

**Definition 20** (Projection Operator). *A projection operator is a mapping*

$$\pi : H \rightarrow S$$

*that assigns a semantic state to a history.*

**Definition 21** (Derived State). *A derived state is a state  $s \in S$  such that*

$$s = \pi(H)$$

*for some projection operator  $\pi$  and history  $H$ .*

**Proposition 4** (Non-Uniqueness of Projection). *There exist distinct projection operators  $\pi_1, \pi_2$  and a history  $H$  such that*

$$\pi_1(H) \neq \pi_2(H).$$

**Theorem 6** (Non-Existence of Global State). *There exists no projection operator  $\pi^*$  such that for all histories  $H$  and all projection operators  $\pi$ ,*

$$\pi^*(H) = \pi(H).$$

**Proposition 5** (Irreversibility of Projection). *For any nontrivial projection operator  $\pi$ , there exist histories  $H_1 \neq H_2$  such that*

$$\pi(H_1) = \pi(H_2).$$

## E Semantic CAP and Distributed Limitation Theorems

**Definition 22** (Semantic System). *A semantic system is a tuple*

$$\Sigma = (N, S, \mathcal{T}, C, \vdash)$$

where  $N$  is a set of nodes,  $S$  is a set of semantic states,  $\mathcal{T}$  is a set of admissible transformations,  $C$  is a set of semantic constraints, and  $\vdash \subseteq S \times C$  is a satisfaction relation.

**Definition 23** (Local Availability). *A semantic system satisfies local availability if for every node  $n \in N$  and every admissible transformation  $t \in \mathcal{T}$  applicable at  $n$ , the transformation can be applied without waiting for coordination with other nodes.*

**Definition 24** (Global Semantic Consistency). *A semantic system satisfies global semantic consistency if for any nodes  $n_1, n_2 \in N$ , the semantic states  $s_1, s_2$  observable at those nodes satisfy*

$$s_1 = s_2.$$

**Definition 25** (Partition Tolerance). *A semantic system is partition tolerant if, for any partition of the node set  $N = N_1 \cup N_2$  with  $N_1 \cap N_2 = \emptyset$ , the system continues to admit transformations within each subset.*

**Definition 26** (Semantic Constraint Preservation). *A semantic system preserves semantic constraints if for all admissible transformations  $t \in \mathcal{T}$  and all states  $s \in S$  satisfying  $s \vdash c$  for some  $c \in C$ , the transformed state  $t(s)$  also satisfies  $t(s) \vdash c$ .*

**Theorem 7** (Semantic CAP Limitation). *No semantic system can simultaneously satisfy global semantic consistency, local availability, partition tolerance, and semantic constraint preservation.*

**Theorem 8** (Undecidability of Optimal Semantic Merge). *Let  $M$  be a merge procedure that selects, for any pair of semantic states, a merged state minimizing entropy production while preserving all satisfiable constraints. The decision problem of determining whether such a merge exists is undecidable.*

**Proposition 6** (Local Sufficiency). *For a semantic system with bounded interaction radius  $r$ , local semantic consistency within radius  $r$  implies operational coherence.*

## References

- [1] C. A. Hidalgo. *The Infinite Alphabet and the Loss of Knowledge*. Penguin Random House, New York, 2025.

- [2] C. A. Hidalgo and R. Hausmann. The building blocks of economic complexity. *Proceedings of the National Academy of Sciences*, 106(26):10570–10575, 2009.
- [3] R. Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3):183–191, 1961.
- [4] C. H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17(6):525–532, 1973.
- [5] G. E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), 1965.
- [6] M. Polanyi. *The Tacit Dimension*. University of Chicago Press, Chicago, 1966.
- [7] H. A. Simon. The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6):467–482, 1962.
- [8] L. A. Suchman. *Plans and Situated Actions*. Cambridge University Press, Cambridge, 1987.
- [9] K. E. Weick. *Sensemaking in Organizations*. Sage Publications, Thousand Oaks, 1995.
- [10] E. Ostrom. *Governing the Commons*. Cambridge University Press, Cambridge, 1990.
- [11] J. C. Scott. *Seeing Like a State*. Yale University Press, New Haven, 1998.
- [12] F. P. Brooks. No silver bullet: Essence and accidents of software engineering. *Computer*, 20(4):10–19, 1987.
- [13] M. Gell-Mann. *The Quark and the Jaguar*. W.H. Freeman, New York, 1994.
- [14] J. Barandes. Unistochastic quantum theory. *Foundations of Physics*, 51:1–30, 2021.
- [15] F. Guimond. Relativistic scalar–vector plenum theory: Entropy, coherence, and structure without expansion. Preprint, 2024.
- [16] F. Guimond. Spherepop: Event-historical computation and semantic locality. Working manuscript, 2025.