Noun-Free Cognition: Difficulty, Abstraction, and the Mobility of Computation

Flyxion

February 8, 2026

Abstract

Difficulty is commonly treated as an intrinsic property of tasks, problems, or domains, varying only with intelligence, skill, or computational power. This assumption underwrites much of contemporary discourse in artificial intelligence, labor economics, cognitive science, and technological forecasting. This paper argues that the assumption is false. Task difficulty is not an inherent attribute but an emergent relation between a task specification, the set of precompiled affordances available to a system, and the environmental context in which execution occurs. As these elements shift over time, the boundary between what appears easy and what appears hard is continually renegotiated.

We develop a unified framework grounded in the concepts of aspect relegation, abstraction understood as compilation against moving targets, and computational displacement. Within this framework, prompts are generalized beyond linguistic inputs to encompass any boundary condition that partitions the world into resolved and unresolved aspects. Drawing structural parallels to intuitions from computational complexity, Gödelian incompleteness, and assembly theory, we show that all adaptive systems exhibit the same signature behavior: local simplification generates displaced complexity elsewhere, driven by selection pressure toward minimal immediate resistance rather than global optimality. This dynamic explains the persistent failure of difficulty prediction, the hardening of analog burdens under digital ease, and the role of Goodhart dynamics as an evolutionary engine of large-scale technological systems. Technological progress, on this account, is not net simplification but systematic complexity redistribution.

1 Introduction: The Paradox of Predictable Unpredictability

Across domains as varied as artificial intelligence, labor economics, education, and cognitive science, there is a persistent pattern of confident prediction followed by systematic failure. Tasks declared permanently resistant to automation are abruptly rendered trivial, while activities assumed to be straightforward prove stubbornly intractable. These reversals are often treated as empirical surprises or forecasting errors, attributed to insufficient data, poor modeling, or unforeseen breakthroughs. This paper advances a stronger claim. The failures are not incidental. They arise because the underlying concept of difficulty presupposed by such predictions is incoherent.

Difficulty is typically treated as an intrinsic attribute of a task, analogous to weight or size, varying only with the power of the agent attempting it. On this view, intelligence—whether human or artificial—is a capacity that gradually expands to encompass ever more difficult problems. The historical record, however, does not conform to this picture. Instead, it reveals repeated inversions in which tasks long considered benchmarks of intelligence collapse into routine, while seemingly mundane activities resist formalization for decades. These inversions are not anomalies but signatures of a deeper structural instability.

A familiar illustration can be found in the history of chess. For much of the twentieth century, mastery of chess functioned as a proxy for strategic intelligence. The eventual defeat of human champions by specialized machines was initially interpreted as the conquest of a hard cognitive problem. Yet within a short period, chess itself ceased to serve as a meaningful benchmark. The task did not become merely easier; it became strategically irrelevant as a measure of general cognition. What changed was not the game but the scaffolding surrounding it: evaluation functions, search heuristics, hardware, training regimes, and cultural expectations. The apparent difficulty of chess dissolved once its structure aligned with available compilation strategies.

Similar patterns recur in contemporary debates about automation and employment. Experts routinely fail to predict which jobs will be automated next, often expressing surprise when systems excel at tasks involving language, perception, or pattern recognition while struggling with forms of physical manipulation or social coordination that humans perform effortlessly. These surprises persist despite increasing technical sophistication in forecasting models. The reason is not a lack of expertise but a mistaken ontology. Difficulty is not a stable target that can be extrapolated forward. It is a moving boundary produced by historically contingent configurations of tools, abstractions, and environments.

The central thesis of this paper is that difficulty is not a noun but a relation. It emerges at the interface between a task specification, the precompiled structures available to a system, and the constraints imposed by time, energy, coordination, and maintenance. When any of these elements change, the difficulty landscape is recomputed. There is no task whose difficulty remains invariant across contexts, scales, or histories. Treating difficulty as an intrinsic property obscures the mechanisms by which tasks transition between tractable and intractable states.

This relational view has immediate consequences for how cognition itself is understood. If difficulty is not intrinsic, then intelligence cannot be defined as the capacity to solve inherently hard problems. Instead, intelligence consists in the continual renegotiation of what counts as a problem at all. What appears as intuition, skill, or automaticity is better understood as the successful relegation of previously resolved structure into opaque, precompiled form. What appears as deliberation or struggle arises when those compilations fail under changing conditions and internal structure must be re-exposed.

The argument developed here proceeds by removing a series of reifications that structure contemporary discourse. Prompts are generalized beyond linguistic inputs to encompass any boundary condition that partitions the world into given and unresolved aspects. Abstraction is reconceived as compilation rather than representation, operating against environments that continually shift. Predictions of difficulty are shown to fail for the same structural reasons that complexity classifications in computation depend on encoding choices and reference frames. Technological progress is analyzed not as net simplification but as systematic displacement of complexity across layers, with Goodhart dynamics acting as the evolutionary engine of scaling systems.

By treating cognition without nouns—by focusing on operations, gradients, and processes rather than faculties and properties—the paper aims to explain not only why difficulty is unstable, but why that instability is unavoidable. The paradox of predictable unpredictability is resolved once difficulty is understood not as something to be measured in advance, but as something continually produced by the interaction between systems and their environments. In the sections that follow, this framework is developed in detail, beginning with a reconceptualization of the prompt as a general boundary condition rather than a mere input.

2 Prompt as Boundary Condition: Generalizing Input

The term *prompt* has recently acquired a narrow and misleading association with linguistic inputs supplied to artificial intelligence systems. In this restricted usage, a prompt is treated as a string of text that elicits a response from a model, and the difficulty of a task is implicitly attributed to the semantic or syntactic properties of that string. This paper adopts a more general and more precise conception. A prompt is any specification that partitions the world into what is treated as given and what is treated as unresolved. Under this definition, prompts are not confined to language, nor even to symbolic representation. They are boundary conditions imposed on a system that delimit the scope of active resolution.

Seen in this light, a sentence, a book, a diagram, a melody, a physical jig, a bureaucratic form, or a social norm all function as prompts in the same structural sense. Each declares, often implicitly, that certain aspects of the environment have already been accounted for and that attention should be focused elsewhere. A blueprint relegates questions of geometry and proportion to its markings so that construction effort can be directed toward execution. A musical score relegates questions of pitch and rhythm so that performance can proceed without recomposition. A job description relegates questions of purpose and scope so that labor can be coordinated without renegotiating goals at every step. In each case, the prompt does not merely convey information; it reshapes the computational surface presented to the system engaging with it.

This generalization has an important consequence. Difficulty does not attach to the underlying task independently of the prompt through which it is presented. The same underlying activity can appear trivial or impossible depending on which aspects are relegated by the prompt and which are left exposed. A diagram may render a spatial reasoning problem immediately solvable that would be opaque if described verbally. Conversely, a verbal abstraction may collapse a complex visual pattern into a single manipulable concept. The task has not changed, but the boundary between resolved and unresolved structure has been redrawn.

Prompts therefore function as selectors of precompiled affordances. Any system, whether biological, social, or artificial, carries with it a repertoire of cached structures: habits, skills, tools, conventions, algorithms, and institutions. A prompt determines which of these structures can be recruited and which remain inaccessible. When a prompt aligns well with existing compilations, execution appears effortless. When it misaligns, the system is forced to expose internal dependencies that had previously been hidden, and the same task suddenly appears difficult.

This perspective undermines the notion that difficulty can be attributed solely to task specification. Let a task be defined abstractly, independent of presentation. The difficulty experienced by a system depends not only on the task itself, but on the prompt through which the task is instantiated and the environment in which that prompt operates. Changing the prompt changes the effective task by altering the interface between the system's internal structure and the external world. There is therefore no privileged, prompt-independent notion of task difficulty.

Formally, difficulty can be understood as a function of at least four variables: the task to be accomplished, the system attempting it, the prompt that structures the interaction, and the environment that constrains execution. Even holding the task and system fixed, variations in the prompt or environment can produce dramatic shifts in apparent difficulty. This dependence explains why trivial changes in framing, tooling, or representation can lead to disproportionate changes in performance, and why such changes are often misinterpreted as evidence of sudden increases in intelligence or capability.

Once prompts are understood as boundary conditions rather than messages, it becomes clear that they play a constitutive role in cognition. They do not merely initiate processes; they define what counts as a processable problem in the first place. The act of prompting is an act of relegation, pushing certain concerns out of active consideration so that others may be addressed. Cognition, on this view, is inseparable from the continual reconfiguration of these boundaries.

This reconceptualization prepares the ground for a deeper claim. If prompts determine which structures are treated as resolved, then abstraction itself can be understood as the systematic creation of prompts that stabilize such relegations. The next section develops this idea by treating abstraction not as representation, but as compilation against environments that are themselves in motion.

3 Abstraction as Compilation Against Moving Targets

Abstraction is commonly described as the act of discarding detail in order to reveal underlying structure. In cognitive science, it is often framed as the formation of representations that capture essential features while ignoring irrelevant variation.

While this description captures an important phenomenological aspect of abstraction, it obscures its operational role. From the perspective developed here, abstraction is better understood as a form of compilation: a process by which tightly coupled dependencies are resolved and pushed inward, producing an interface that can be treated as opaque for the purposes of further action.

In this sense, abstraction does not eliminate complexity; it relocates it. When a set of interactions is sufficiently stabilized, the system ceases to model its internal dynamics explicitly and instead interacts with it through a simplified surface. A driver presses a pedal rather than computing combustion dynamics. A mathematician invokes a theorem rather than rederiving it. A programmer calls a library function rather than reimplementing an algorithm. In each case, abstraction functions as a compiled artifact that compresses prior effort into a reusable form.

This compilation, however, differs in a crucial respect from compilation in formal computing systems. In software, compilation typically occurs against a fixed target environment. The semantics of the source language, the architecture of the machine, and the operating conditions are assumed to be stable. Once compiled, a program can be executed repeatedly with predictable behavior. Cognitive and social compilation, by contrast, occurs against environments that are continually shifting. Tools evolve, contexts change, constraints fluctuate, and interactions accumulate. What was once a stable abstraction may become brittle or misleading as the conditions under which it was compiled drift.

As a result, abstraction is inherently provisional. The interfaces it produces are only reliable so long as the environmental assumptions that underwrote their compilation remain approximately valid. When those assumptions fail, the abstraction must be dismantled, and its internal structure re-exposed to active reasoning. This re-exposure is often experienced subjectively as difficulty, confusion, or error, and objectively as breakdowns in coordination, performance, or prediction. The task has not grown more complex in any absolute sense; the abstraction has lost its alignment with the world.

This dynamic explains why expertise is both powerful and fragile. Experts operate by leveraging deeply compiled structures that allow them to bypass enormous combinatorial spaces. Their apparent ease reflects the extent to which relevant dependencies have been relegated beneath conscious access. Yet this same relegation renders them vulnerable to context shifts that invalidate their compilations. Novices, lacking such abstractions, may perform worse under stable conditions but adapt more readily when those conditions change, precisely because fewer assumptions are hidden from view.

The instability of abstraction can be stated as an informal theorem. No compilation strategy remains optimal across all environments. Any abstraction that reduces immediate cognitive or computational load does so by embedding assumptions about regularities in the world. As those regularities change, the abstraction accumulates error, eventually requiring recompilation. Because environments are open-ended and historically contingent, there is no final compilation that secures permanent ease.

This instability undermines the search for intrinsic measures of task difficulty. A task appears easy when appropriate abstractions are already compiled and aligned with current conditions. It appears hard when such abstractions are absent, misaligned, or in the process of being dismantled. Difficulty is therefore not a property of the task but a symptom of a mismatch between compiled structure and present demands.

Understanding abstraction as compilation against moving targets also clarifies the relationship between cognition and environment. Cognitive processes do not merely operate within a world; they coevolve with it. Each act of abstraction reshapes the environment by enabling new forms of action, coordination, and exploitation. These changes, in turn, alter the conditions under which existing abstractions were compiled, accelerating their obsolescence. The result is a feedback loop in which abstraction both stabilizes and destabilizes cognition over time.

The consequences of this view extend beyond individual cognition to collective and technological systems. When abstractions are externalized into tools, standards, and institutions, their provisional nature becomes harder to recognize. Breakdowns are then interpreted as failures of agents rather than as signals that the compilation boundary itself has shifted. The next section shows how this misrecognition contributes to systematic prediction failures by obscuring the scale-relative and history-dependent nature of complexity.

4 Persistent Asymmetries and the Limits of Metaphor

One consequence of the framework developed so far is that there will always exist tasks that are easy for humans and difficult for machines, just as there will always exist tasks that are easy for machines and difficult for humans. This is not a transient gap to be closed by progress, nor an artifact of incomplete engineering, but a structural feature of systems whose compilations evolve along different historical and environmental trajectories. The asymmetry is not located in intelligence itself but in the kinds of structure each system has already relegated, stabilized, and externalized.

For this reason, the standard use of chess as a paradigmatic benchmark of intelligence is misleading in more ways than one. Chess is not merely a bounded combinatorial game with fixed rules; it is a sociohistorical artifact whose significance depends on cultural attention, institutional reinforcement, and shared valuation. Its rise as a measure of intelligence and its subsequent decline as a meaningful benchmark were driven as much by shifts in prestige, pedagogy, and professional relevance as by advances in computation. Once chess engines surpassed human play, the game did not simply become easy; it lost its role as a site where difficulty mattered. The metaphor therefore conflates the resolution of a technical challenge with the erosion of the social conditions that made that challenge salient in the first place.

A more revealing prototype is provided by games such as Tetris. Unlike chess, Tetris does not derive its difficulty from deep strategic foresight or symbolic reasoning, but from real-time perception, motor coordination, and the continuous management of spatial constraints under time pressure. Its challenge is inseparable from embodiment, attentional rhythm, and perceptual grouping. Humans find such tasks natural not because they are simple in an abstract sense, but because they align closely with compilations produced by millions of years of sensorimotor evolution. Machines, by contrast, must explicitly construct representations and control policies that approximate what humans already possess in relegated form.

The significance of Tetris as a prototype lies in its resistance to stable abstraction. There is no compact symbolic summary that captures the full dynamics of play in a way that renders execution trivial. Each piece arrives with local contingencies that must be resolved in real time, and small errors propagate rapidly into failure. For humans, this aligns with deeply compiled perceptual and motor loops that operate beneath conscious deliberation. For machines, the same task requires explicit modeling of state transitions, reward structures, and timing constraints, often at a computational cost disproportionate to its apparent simplicity.

This contrast illustrates why there can be no final convergence in which machines simply absorb all domains of human ease. Human ease is not defined by problem domains but by alignment between tasks and embodied compilations. As machines acquire new compilations, new asymmetries will emerge, because the environments and pressures that shape those compilations differ. What becomes trivial for machines will often be what can be cleanly externalized into stable abstractions, while what remains difficult will be precisely what resists such stabilization.

The broader implication is that difficulty landscapes are not merely mobile over time but plural across systems. A task's apparent simplicity for one agent does not imply its simplicity for another, even when both are highly capable. This plurality further undermines the notion of a single, objective hierarchy of task difficulty. It also cautions against the use of culturally contingent benchmarks as proxies for intelligence or progress.

By shifting attention away from celebrated symbolic victories and toward mundane, continuous, and embodied activities, the analysis reinforces the central claim of noun-free cognition. Difficulty does not reside in tasks as such, nor does ease signal intrinsic superiority. Both are outcomes of historical compilation paths interacting with present constraints. As those paths diverge, asymmetries persist, not as anomalies to be eliminated, but as inevitable features of adaptive systems operating under different conditions.

This perspective sets the stage for a more general explanation of why prediction failures recur even when asymmetries are acknowledged. To understand that persistence, it is necessary to examine how complexity itself depends on scale and history, and why classifications that appear stable at one level collapse when viewed from another. The next section develops this argument by generalizing intuitions from computational complexity and formal incompleteness to adaptive systems more broadly.

5 Why Predictions Fail: Scale, History, and the Relativity of Complexity

The persistent failure of difficulty prediction follows directly from the instability of compilation boundaries. Forecasts implicitly assume that the structures which currently render a task easy or hard will remain intact as other variables change. This assumption is rarely justified. In adaptive systems, complexity is neither absolute nor monotonic; it is scale-relative and history-dependent. What appears tractable at one level of description may be intractable at another, and what appears stable over one historical window may collapse entirely over the next.

This relativity mirrors a familiar intuition from computational complexity theory, where the classification of problems depends on the resources permitted and the encoding chosen. Whether a problem is considered efficiently solvable is inseparable from assumptions about what counts as a basic operation. In lived systems, those assumptions correspond to what has already been compiled into tools, skills, or infrastructure. A task becomes easy not because its combinatorics have vanished, but because they have been absorbed into a substrate that is no longer interrogated.

Historical path dependence intensifies this effect. Once a particular compilation strategy becomes dominant, it reshapes the environment in which future tasks are encountered. New tools encourage new forms of problem specification, which in turn privilege certain abstractions over others. Over time, entire difficulty landscapes are reorganized around contingencies that appear natural only in retrospect. Predictions that extrapolate from the present without accounting for this recursive restructuring inevitably fail, not because the future is opaque, but because the act of simplification itself changes the conditions being projected.

There is also a structural parallel to formal incompleteness. Any attempt to define a comprehensive measure of difficulty presupposes a fixed frame of reference. Yet adaptive systems continually modify the very resources upon which such measures depend. Just as no sufficiently expressive formal system can fully capture its own behavior, no difficulty metric can remain valid across the transformations induced by its own use. As soon as a measure becomes salient, it becomes a target for optimization, and its classificatory power erodes.

Empirical reversals illustrate this point repeatedly. Tasks long assumed to require deep intelligence collapse once appropriate scaffolding is introduced, while activities dismissed as trivial resist automation because they rely on compilations that are difficult to externalize. These reversals are not contradictions but manifestations of the same principle: complexity does not reside in tasks but in mismatches between tasks and available structure. Prediction fails whenever that structure shifts in ways that were not, and often could not be, anticipated.

Recognizing the scale-relative nature of complexity also clarifies why expert judgment offers limited protection against error. Expertise consists in mastery of existing compilations, not foresight into their obsolescence. Experts project forward from current boundaries, mistaking historically contingent scaffolding for intrinsic structure. When those boundaries dissolve, expertise appears to fail, though in reality the frame of reference has changed beneath it.

If prediction failure arises from the instability of compilation boundaries, then technological progress must be reinterpreted accordingly. Rather than eliminating difficulty, new technologies redistribute it across layers of activity. Understanding this redistribution requires examining where complexity goes when it appears to disappear. The next section addresses this question by analyzing computational displacement and the hardening of analog burdens under digital ease.

6 Computational Displacement and the Hardening of the Analog World

Technological progress is often described as a process of simplification. Tasks that once demanded skill, time, or coordination are rendered effortless through automation, digitization, or standardization. This narrative, while locally accurate, is globally misleading. Simplification at one layer is achieved by displacing complexity to another. What disappears from the user-facing surface reappears as infrastructure, maintenance, coordination, and constraint elsewhere in the system.

When analytical or symbolic processes are compressed into software, the immediate cognitive burden on users is reduced. Yet this reduction is contingent on an expanding substrate of physical and organizational support. Computation requires energy, cooling, fabrication, logistics, and continual upkeep. Digital artifacts demand version control, compatibility management, security, and repair. The apparent immateriality of software conceals a dense material and social scaffolding that must be sustained for the simplification to persist.

This displacement also reshapes human activity. As tasks become easier to initiate, they proliferate. Reduced friction invites increased volume, which in turn generates new forms of overload. Attention becomes fragmented, coordination costs rise, and the effort required to maintain coherence across proliferating processes increases. The ease of execution produces difficulty of management. What was once hard to do becomes hard to stop, regulate, or integrate.

The hardening of analog burdens is a particularly striking consequence. Interfaces optimized for symbolic manipulation impose repetitive physical actions, sustained screen time, and constrained postures. Maintenance labor shifts from visible craftsmanship to invisible troubleshooting. Errors become harder to detect as systems grow more opaque. The labor saved in one domain reappears as strain, fatigue, and vigilance in another, often distributed unevenly across populations.

These effects are not accidental side consequences but structural necessities. Complexity is not destroyed by computation; it is reorganized. The system as a whole may even become more complex as simplifications enable denser interconnections and tighter coupling. From the perspective of any single layer, progress appears undeniable. From the perspective of the whole, entanglement increases.

This conservation-like behavior undermines accounts of progress that focus solely on efficiency gains. Measuring what has become easier without tracking what has become harder yields a distorted picture. The difficulty that vanishes from one surface does not evaporate; it migrates, often to places that are less visible, less prestigious, or less easily quantified.

Computational displacement also sets the stage for a further dynamic. Once simplifications are codified into metrics and targets, they invite optimization that amplifies displacement rather than stabilizing it. This dynamic, commonly discussed under the heading of Goodhart's Law, functions as an evolutionary engine that drives systems toward ever greater scale and complexity. The following section examines this process in detail.

7 Goodhart Dynamics and the Reification of Process

The pressure to convert fluid practices into stable objects is not merely a conceptual temptation; it is a structural feature of large-scale coordination. Systems that must allocate resources, evaluate performance, or govern behavior tend to replace open-ended processes with metrics that can be counted, compared, and optimized. This substitution appears innocuous, even necessary, but it initiates a dynamic that systematically undermines the very stability it seeks to impose. The phenomenon commonly summarized as Goodhart's Law captures this dynamic in compressed form, but its implications extend far beyond measurement error.

When a process is reified into a metric, it becomes tractable to optimize. Optimization, in turn, invites strategic behavior that exploits the gap between the proxy and the practice it stands in for. As actors learn to satisfy the metric with minimal immediate effort, the underlying process is distorted. Compensatory structures are then introduced to repair the damage: additional rules, oversight mechanisms, secondary metrics, or technological fixes. Each layer of correction increases systemic complexity while narrowing the range of contexts in which the metric remains meaningful.

This cycle does not represent a failure of governance or design; it is the predictable outcome of treating dynamic relations as nouns. Once difficulty, productivity, intelligence, or quality are frozen into targets, they cease to function as indicators and become sites of selection pressure. The system adapts to the measurement rather than to the phenomenon measured. What follows is not equilibrium but escalation, as each attempt to stabilize the abstraction generates new avenues for divergence.

This dynamic was first articulated in economic contexts by Goodhart, but its scope is general. In educational systems, test scores replace learning and provoke teaching to the test. In scientific research, citation counts replace insight and incentivize salami slicing and fashionable conformity. In software development, performance benchmarks replace robustness and yield brittle optimizations that collapse outside narrow conditions. In each case, a relational process is converted into a noun-like quantity, optimized locally, and rendered unstable globally.

From the perspective of noun-free cognition, Goodhart dynamics are not pathologies to be corrected but signals of a deeper mismatch. They arise whenever a system attempts to fix meaning where meaning must remain flexible. Metrics function as compiled abstractions that temporarily reduce coordination cost, but their success guarantees their eventual failure by altering the environment they summarize. As optimization proceeds, the assumptions embedded in the metric are violated, and the abstraction must either be revised or replaced.

This explains why large technical systems tend to grow more complex over time despite repeated simplification efforts. Each successful abstraction becomes a new site of exploitation, requiring further abstraction to contain it. The resulting structure resembles an evolutionary arms race rather than a march toward efficiency. Local gains accumulate into global entanglement, and the apparent clarity of metrics conceals a proliferation of hidden dependencies.

The same logic applies to technological scaling. Narratives of progress often emphasize exponential improvements in speed, capacity, or cost, treating these curves as evidence of increasing mastery. Yet such curves track only what has been rendered legible to measurement. Beneath them lies a growing mass of unmeasured work: fabrication complexity, energy expenditure, environmental impact, maintenance labor, and coordination overhead. The metric improves precisely because these costs have been displaced rather than eliminated.

Seen in this light, Goodhart dynamics are the operational mechanism by which complexity displacement accelerates. Each time a relational process is nounified and optimized, it creates conditions that demand further nounification elsewhere. The system becomes increasingly reliant on abstractions whose validity windows shrink even as their influence expands. Prediction becomes harder not because the world is chaotic, but because the act of measuring and optimizing reshapes the terrain faster than models can track.

This analysis returns the discussion to the central claim of the paper. Difficulty is not an object to be minimized but a relation to be managed. Attempts to eliminate it through metric optimization merely move it into less visible forms, where it reappears as fragility, overload, or breakdown. To treat difficulty as a noun is therefore to guarantee its recurrence in displaced form.

The remaining task is to situate this dynamic within a more general account of adaptive behavior. If systems consistently follow paths that minimize immediate resistance, even at the cost of long-term entanglement, then Goodhart dynamics are not accidental deviations but expressions of a deeper gradient.

8 Language Games, Family Resemblance, and the Resistance to Nounification

The framework developed in this paper can be read as an explicit generalization of a set of insights that were already present, though not systematized, in the later philosophy of *Ludwig Wittgenstein*. In his investigations of language, Wittgenstein rejected the assumption that meaning is grounded in stable definitions or abstract essences. Instead, he emphasized use, practice, and context, introducing the notions of language games and family resemblance to describe how words function without fixed boundaries. These notions were deliberately left underdefined, not as a failure of rigor, but as a methodological refusal to reify what is inherently dynamic.

Language games designate patterns of activity in which words acquire meaning through participation rather than correspondence. To understand a term is not to grasp a definition but to know how it is used across a range of situations. This knowledge is practical and distributed, embedded in forms of life rather than encoded in explicit rules. As Wittgenstein repeatedly emphasized, speakers do not consult a definition each time they speak; they act within a practice whose contours are continually renegotiated. Meaning, on this view, is not a property of words but an emergent regularity of coordinated activity.

Family resemblance extends this insight by denying that categories must be unified by a single shared feature. Members of a category may overlap in multiple, crisscrossing ways without any invariant core. Wittgenstein's canonical examples deliberately resist summarization because any attempt to extract a common essence misrepresents how the category actually functions. What holds the category together is not a definition but a pattern of resemblances sustained through use.

These ideas align directly with the rejection of nounification advanced in this paper. Treating difficulty, intelligence, or cognition as nouns presupposes that they name stable objects or properties. Wittgenstein's analysis shows why such presuppositions fail even in the seemingly simpler case of ordinary language. If the meaning of a word is inseparable from its use, and if that use shifts across contexts, then the referent of the word cannot be fixed without distorting the practice it describes.

From the present perspective, language games can be understood as historically evolved compilations that stabilize certain patterns of interaction while leaving others open. Each use of a word implicitly selects a subset of prior uses as relevant and relegates others to the background. In doing so, it redraws the boundary of the game itself. Meaning is not merely applied; it is recomputed. To speak is therefore to participate in a process of continual recompilation, in which previously stabilized abstractions are reaffirmed, modified, or dissolved.

This explains why Wittgenstein resisted formal definition. Definitions aim to freeze meaning by specifying necessary and sufficient conditions. Language games and family resemblances, by contrast, remain functional precisely because they do not settle into a single form. Their flexibility allows them to track shifting practices and environments. Attempting to define them would not clarify their operation but undermine it by imposing artificial stability.

The same logic applies beyond language. Just as words derive meaning from use within changing practices, tasks derive difficulty from their embedding in evolving scaffolds. Each time a task is performed, the surrounding context is slightly altered: tools are refined, habits are reinforced or broken, expectations shift. In this sense, every act of cognition redefines the language in which it operates. There is no final vocabulary in which tasks can be classified once and for all.

Seen through this lens, noun-free cognition is not a radical departure from established philosophy but a continuation of Wittgenstein's anti-essentialist program. What he demonstrated for words, this paper extends to difficulty, abstraction, and intelligence. The refusal to define these terms is not a gesture of obscurity but a recognition that their apparent stability is a byproduct of temporarily successful coordination. Their meaning, like that of language itself, exists only in motion.

This connection also clarifies why attempts to formalize intelligence or difficulty inevitably trail practice rather than govern it. Formalizations arise as retrospective summaries of stabilized use. They are useful so long as the practices they summarize remain intact, and they fail when those practices shift. Wittgenstein's insistence that "we know what we mean" without being able to state it captures this asymmetry. Knowledge precedes definition, and use precedes theory.

By making explicit the structural parallels between language games and computational scaffolding, the argument situates noun-free cognition within a broader philosophical tradition that treats meaning, difficulty, and understanding as relational achievements rather than static entities. The remaining sections extend this analysis to metric optimization and Goodhart dynamics, showing how attempts to force noun-like stability onto fluid practices generate the very instabilities they seek to eliminate.

9 Assembly Index Minimization and the Gradient of Immediate Resistance

The dynamics described thus far can be unified under a more general principle governing adaptive systems: the tendency to minimize immediate resistance rather than total cost. This tendency appears across physical, biological, cognitive, and social domains, and it provides a common explanation for why local simplification so reliably produces global entanglement. Assembly theory offers a useful formal lens through which to articulate this principle without reintroducing the nounification the present framework seeks to avoid.

In assembly theory, the complexity of an object is characterized by its assembly index, defined as the minimal number of steps required to construct that object from a given set of basic components. Objects with low assembly index are readily produced through simple processes, while objects with high assembly index require extended sequences of coordinated actions. Importantly, assembly index is not an intrinsic measure in isolation; it is defined relative to an available toolkit and a history of prior constructions. What counts as a basic component at one moment may itself be a high-level composite at another.

When viewed dynamically, adaptive systems tend to follow trajectories that minimize the immediate increase in assembly index at each step. They do not search for globally minimal constructions but exploit locally available shortcuts. A chemical reaction proceeds along the path of lowest activation energy, not toward the deepest global energy minimum. Biological evolution selects for local fitness improvements, even when those improvements constrain future adaptation. Cognitive agents favor heuristics that reduce immediate cognitive load, even when those heuristics accumulate bias or fragility over time.

This principle extends naturally to abstraction and computation. A compiled abstraction reduces the apparent assembly index of future actions by embedding prior work into a reusable form. From the system's perspective, this represents an immediate reduction in resistance. The fact that such abstractions impose future costs by constraining flexibility or increasing coupling is secondary, because those costs are deferred and often externalized. The gradient that is followed is not total complexity minimization but next-step ease.

Once this asymmetry is recognized, the inevitability of complexity displacement becomes clear. Each local reduction in assembly index reshapes the landscape, making new constructions possible that were previously inaccessible. These constructions, in turn, introduce dependencies that raise the assembly index of the system as a whole. What appears as progress from within the local frame manifests as escalating coordination cost when viewed from a broader scale.

This gradient also explains why attempts to stabilize systems through optimization tend to backfire. Optimization selects for strategies that minimize immediate cost under current metrics. As those strategies proliferate, they alter the environment in ways that invalidate the assumptions of the optimization itself. New layers of abstraction are introduced to manage the resulting complexity, each following the same gradient. The system never converges on a stable optimum because the act of optimization continually reshapes the space being optimized.

The connection to thermodynamics is suggestive but limited. In physical systems, local decreases in entropy are permitted so long as entropy is exported elsewhere. In cognitive and social systems, local decreases in apparent complexity are permitted so long as complexity is displaced into infrastructure, coordination, or future maintenance. The analogy is not exact, but the structural similarity is instructive. In both cases, order is achieved locally by reorganizing rather than eliminating constraint.

Understanding assembly index minimization as a general adaptive principle allows us to reinterpret gravity, self-organization, and learning without invoking separate explanatory frameworks. Gravity follows the steepest descent permitted by spacetime geometry. Self-organizing systems settle into configurations that locally dissipate free energy. Learning systems compress representations to reduce immediate prediction error. In each case, the system exploits the most readily available path, even if that path increases long-term rigidity or coupling.

This perspective dissolves the expectation that increasing intelligence or capability should lead to globally simpler systems. On the contrary, greater capability expands the space of possible shortcuts, accelerating the redistribution of complexity. The more effectively a system can minimize immediate resistance, the more rapidly it generates higher-order structure that demands further management.

The final sections of the paper draw out the implications of this analysis. If difficulty is mobile, relational, and continually displaced, then intelligence must be redefined accordingly. What follows is not a proposal for eliminating difficulty, but a reframing of what it means to navigate it responsibly in cognitive, technological, and social systems.

10 Assembly Index Minimization and the Gradient of Immediate Resistance

The dynamics described thus far can be unified under a more general principle governing adaptive systems: the tendency to minimize immediate resistance rather than total cost. This tendency appears across physical, biological, cognitive, and social domains, and it provides a common explanation for why local simplification so reliably produces global entanglement. Assembly theory offers a useful formal lens through which to articulate this principle without reintroducing the nounification the present framework seeks to avoid.

In assembly theory, the complexity of an object is characterized by its assembly index, defined as the minimal number of steps required to construct that object from a given set of basic components. Objects with low assembly index are readily produced through simple processes, while objects with high assembly index require extended sequences of coordinated actions. Importantly, assembly index is not an intrinsic measure in isolation; it is defined relative to an available toolkit and a history of prior constructions. What counts as a basic component at one moment may itself be a high-level composite at another.

When viewed dynamically, adaptive systems tend to follow trajectories that minimize the immediate increase in assembly index at each step. They do not search for globally minimal constructions but exploit locally available shortcuts. A chemical reaction proceeds along the path of lowest activation energy, not toward the deepest global energy minimum. Biological evolution selects for local fitness improvements, even when those improvements constrain future adaptation. Cognitive agents favor heuristics that reduce immediate cognitive load, even when those heuristics accumulate bias or fragility over time.

This principle extends naturally to abstraction and computation. A compiled abstraction reduces the apparent assembly index of future actions by embedding prior work into a reusable form. From the system's perspective, this represents an immediate reduction in resistance. The fact that such abstractions impose future costs by constraining flexibility or increasing coupling is secondary, because those costs are deferred and often externalized. The gradient that is followed is not total complexity minimization but next-step ease.

Once this asymmetry is recognized, the inevitability of complexity displacement becomes clear. Each local reduction in assembly index reshapes the landscape, making new constructions possible that were previously inaccessible. These constructions, in turn, introduce dependencies that raise the assembly index of the system as a whole. What appears as progress from within the local frame manifests as escalating coordination cost when viewed from a broader scale.

This gradient also explains why attempts to stabilize systems through optimization tend to backfire. Optimization selects for strategies that minimize immediate cost under current metrics. As those strategies proliferate, they alter the environment in ways that invalidate the assumptions of the optimization itself. New layers of abstraction are introduced to manage the resulting complexity, each following the same gradient. The system never converges on a stable optimum because the act of optimization continually reshapes the space being optimized.

The connection to thermodynamics is suggestive but limited. In physical systems, local decreases in entropy are permitted so long as entropy is exported elsewhere. In cognitive and social systems, local decreases in apparent complexity are permitted so long as complexity is displaced into infrastructure, coordination, or future maintenance. The analogy is not exact, but the structural similarity is instructive. In both cases, order is achieved locally by reorganizing rather than eliminating constraint.

Understanding assembly index minimization as a general adaptive principle allows us to reinterpret gravity, self-organization, and learning without invoking separate explanatory frameworks. Gravity follows the steepest descent permitted by spacetime geometry. Self-organizing systems settle into configurations that locally dissipate free energy. Learning systems compress representations to reduce immediate prediction error. In each case, the system exploits the most readily available path, even if that path increases long-term rigidity or coupling.

This perspective dissolves the expectation that increasing intelligence or capability should lead to globally simpler systems. On the contrary, greater capability expands the space of possible shortcuts, accelerating the redistribution of complexity. The more effectively a system can minimize immediate resistance, the more rapidly it generates higher-order structure that demands further management.

The final sections of the paper draw out the implications of this analysis. If difficulty is mobile, relational, and continually displaced, then intelligence must be redefined accordingly. What follows is not a proposal for eliminating difficulty, but a reframing of what it means to navigate it responsibly in cognitive, technological, and social systems.

11 Implications for Cognition, Technology, and Forecasting

If difficulty is not an intrinsic property but a relational outcome of compilation histories interacting with present constraints, then several dominant assumptions across cognitive science, artificial intelligence, and technology policy must be revised. Intelligence can no longer be coherently defined as the capacity to solve inherently hard problems, nor can progress be measured by the steady conquest of difficulty. What appears instead is a picture of continual renegotiation, in which systems succeed by reorganizing what counts as effort rather than by eliminating it.

For cognitive science, this implies that distinctions commonly treated as architectural, such as the separation between automatic and deliberative processes, should be understood as differences in compilation depth rather than as distinct faculties. Processes experienced as intuitive or effortless are those whose internal dependencies have been relegated beneath conscious access through prior stabilization. Processes experienced as difficult or effortful are those in which such relegation has failed or been temporarily reversed. Cognition, on this view, is not divided into systems but modulated in resolution, with attention functioning as a mechanism for exposing or concealing structure as circumstances demand.

In the context of artificial intelligence, the framework cautions against extrapolations that treat current benchmarks as stable indicators of general capability. Questions framed as predictions about when a system will "solve" a given task presuppose that the task has a fixed identity and that its difficulty can be projected independently of the scaffolding that renders it tractable. In practice, advances in artificial systems frequently succeed by altering task definitions, representations, and interfaces, thereby collapsing difficulty rather than confronting it directly. Forecasts fail because they ignore the coevolution of tasks and tools.

Technological policy is similarly affected. Automation is often evaluated in terms of efficiency gains or job displacement, with insufficient attention to where complexity migrates. When systems are designed to reduce friction at the point of use, they typically increase complexity in infrastructure, regulation, and maintenance. Policies that focus solely on visible gains risk underestimating the long-term costs imposed on coordination, resilience, and human well-being. Managing technological change therefore requires tracking displaced complexity, not merely counting outputs.

Forecasting more generally must contend with the reflexive nature of simplification. Predictions alter incentives, incentives reshape behavior, and behavior reshapes the environment in which predictions were made. This reflexivity ensures that difficulty landscapes shift in response to the very models used to describe them. Accurate long-term prediction is therefore not merely difficult but structurally constrained. The appropriate object of forecasting is not which tasks will become easy or hard, but how compilation strategies and displacement patterns are likely to evolve.

These implications converge on a redefinition of intelligence itself. Intelligence is not best conceived as a store of representations or a scalar capacity, but as an ongoing process of boundary management. Intelligent systems are those that can detect when existing abstractions have become misaligned and can reconfigure their compilations accordingly. Stupidity, in this sense, is not a lack of capability but an overcommitment to obsolete abstractions that no longer fit the environment.

This redefinition has ethical consequences. If difficulty is relational, then fairness cannot be assessed solely by comparing outcomes across individuals or groups. Different agents operate with different compilations, shaped by unequal access to tools, training, and supportive environments. What appears as failure or laziness from one perspective may reflect a mismatch between imposed prompts and available scaffolding. Ethical evaluation must therefore attend to the distribution of precompiled affordances rather than assuming uniform difficulty.

The paper concludes by returning to its deflationary stance. Nothing in this framework requires postulating hidden depths or ineffable properties. Difficulty appears mysterious only when treated as a thing. Once reinterpreted as a process, its mobility becomes unsurprising. The remaining task is not to eliminate difficulty, but to design systems that can adaptively renegotiate it without allowing displacement to accumulate beyond manageable bounds. The concluding section draws these threads together and articulates the positive program implied by noun-free cognition.

12 Conclusion: Difficulty Without Nouns

This paper has argued that difficulty should not be treated as a noun, an intrinsic attribute of tasks, domains, or problems. It is instead a relational phenomenon produced by the interaction between task specifications, compiled structures, and environmental constraints. Because each of these elements evolves over time, the boundary between ease and effort is inherently unstable. What appears predictable in one context becomes opaque in another, not because of error or ignorance, but because the object of prediction has changed.

By reconceptualizing prompts as boundary conditions and abstraction as compilation against moving targets, the analysis dissolves the intuition that progress consists in the steady elimination of hard problems. Simplification occurs, but it does so locally, displacing complexity into infrastructure, coordination, and future maintenance. Goodhart dynamics ensure that any attempt to stabilize such simplifications through metrics accelerates this displacement, while assembly index minimization explains why systems consistently follow paths of least immediate resistance even when those paths generate long-term entanglement.

The resulting picture is neither pessimistic nor utopian. It does not deny the reality of improvement, nor does it promise a final equilibrium. Instead, it reframes improvement as a matter of redistribution rather than conquest. Intelligence, on this account, is the capacity to navigate shifting boundaries of difficulty by revising abstractions as contexts change. Failure arises not from insufficient power but from rigidity in the face of such change.

The positive program implied by noun-free cognition is therefore pragmatic rather than heroic. Rather than seeking to solve hard problems once and for all, systems should be designed to monitor where complexity accumulates, to detect when abstractions have become brittle, and to facilitate recompilation without catastrophic breakdown. Such systems will not eliminate difficulty, but they may prevent its displacement from becoming destructive.

Perhaps the most general lesson is that any system capable of simplifying its own processes inevitably reshapes its environment in ways that generate new forms of difficulty. This is not a flaw to be corrected but a condition to be acknowledged. The question is not whether difficulty can be abolished, but whether its continual reappearance can be managed with foresight, humility, and care.

Appendices

A Formal Sketch of Relational Difficulty

This appendix provides a minimal mathematical sketch of the relational conception of difficulty used throughout the paper. The purpose is not to propose a finalized formalism, but to show that the central claims can be expressed without contradiction in a precise idiom.

Let T denote a task specification, understood abstractly as a mapping from initial conditions to desired outcomes. Let S denote a system attempting to perform the task, characterized by its internal state, available operations, and precompiled structures. Let $\mathcal{P}$ denote the set of compiled affordances available to S, including tools, habits, algorithms, and external scaffolds. Let E denote the environment, including temporal constraints, energy availability, coordination costs, and background regularities.

We define difficulty as a function

$$D : (T, S, \mathcal{P}, E) \longrightarrow \mathbb{R}^+,$$

where D measures the expected cost of successful execution under the given configuration. Here $\mathbb{R}^+$ denotes the set of non-negative real numbers, reflecting that difficulty corresponds to a contragrade process requiring some expenditure of effort, rather than an orthograde or spontaneous process that proceeds without work. Crucially, D is not invariant under transformations of $\mathcal{P}$ or E, even when T and S are held fixed. Small perturbations to $\mathcal{P}$, such as the introduction of a new abstraction or tool, may produce discontinuous changes in D.

To formalize abstraction as compilation, let $\mathcal{P}_{t+1} = \mathcal{P}_t \cup \{a\}$, where a is a compiled structure produced by resolving a subset of dependencies internal to T. The introduction of a reduces the apparent complexity of subsequent executions by replacing a composite operation with an atomic one. However, this reduction is conditional on environmental assumptions embedded in a. If those assumptions cease to hold, a must be decomposed, effectively reversing the compilation.

This formal dependence implies that no task admits a context-free difficulty measure. For any fixed $D(T)$, there exist environments E_1, E_2 and compilations $\mathcal{P}_1, \mathcal{P}_2$ such that $D(T, S, \mathcal{P}_1, E_1) \ll D(T, S, \mathcal{P}_2, E_2)$. Difficulty is therefore relational and historically contingent, not intrinsic.

B Toy Example: Graph Traversal With and Without Compilation

Consider a graph $G = (V, E)$ with designated start node s and target node t. The task T is to find a path from s to t. For a system S with no precompiled structures, difficulty corresponds to the cost of search, which may scale exponentially with graph size in the worst case.

Now suppose the system acquires a compiled structure a in the form of a cached shortest-path tree rooted at s. The effective task is transformed. Path retrieval becomes a constant-time operation relative to graph size. The difficulty function collapses not because the graph has changed, but because a large portion of its combinatorial structure has been relegated into $\mathcal{P}$.

If the graph is later modified by adding or removing edges, the cached structure a may become invalid. In that case, the system must either accept incorrect paths or re-expose the full graph structure to recompute the cache. The difficulty of the original task re-emerges, not because the task has grown harder, but because the compilation has lost alignment with the environment.

This example illustrates three general points. First, difficulty depends on available compilations. Second, compilations embed environmental assumptions. Third, environmental change can reintroduce previously hidden complexity without altering the nominal task.

C Displacement as Complexity Redistribution

To capture computational displacement, consider total system cost C_{total} decomposed into visible execution cost C_{exec} and background support cost C_{sup}:

$$C_{\text{total}} = C_{\text{exec}} + C_{\text{sup}}.$$

Automation and abstraction typically aim to minimize C_{exec}. However, reductions in C_{exec} are achieved by increasing C_{sup} through infrastructure, maintenance, coordination, and monitoring. The apparent simplification is therefore conditional on a redistribution of cost rather than its elimination.

In many systems, C_{sup} grows superlinearly with scale due to coupling effects and coordination overhead. As a result, global complexity increases even as local execution becomes easier. This explains why large systems often become fragile despite continual local optimization.

The framework does not assert strict conservation of complexity, but it predicts systematic displacement. Simplification at one interface reliably produces entanglement elsewhere, particularly in layers that are less visible to direct measurement or optimization.

D Assembly Index and Immediate Resistance

Let $A(x)$ denote the assembly index of an object or process x relative to a given basis set of primitives. Consider a sequence of constructions $\{x_0, x_1, \ldots, x_n\}$ produced by an adaptive system. At each step, the system selects an operation that minimizes the incremental increase $\Delta A = A(x_{k+1}) - A(x_k)$ subject to current constraints.

This local minimization strategy does not guarantee minimal total assembly index over the sequence. On the contrary, it often leads to configurations with high global assembly index due to accumulated dependencies. Nevertheless, such trajectories are favored because they reduce immediate resistance.

This formalizes the claim that adaptive systems follow gradients of least immediate resistance rather than least total complexity. The same principle applies to abstraction, metric optimization, and institutional evolution. Each step is locally justified, even as the global structure becomes increasingly difficult to manage.

E Limits of Formalization

Finally, it is important to note that any formalization of difficulty inherits the same instability it seeks to describe. Once a measure of difficulty is adopted, it becomes a candidate for optimization, altering the system it measures. This reflexivity mirrors the incompleteness phenomena discussed in the main text. Formal models are therefore best understood as temporary stabilizations rather than definitive accounts.

The mathematical sketches offered here should be read in that spirit. They demonstrate coherence, not closure. They show that noun-free cognition can be expressed formally, but they also reinforce the central claim of the paper: no formal system can freeze a process whose defining feature is continual renegotiation.

F Formal Consequences of Prompt Relativity

This appendix develops a more explicit account of prompt relativity and shows how changes in prompting alone can induce phase-like transitions in apparent task difficulty without any modification to the underlying task or system.

Let a task T be represented as a constraint satisfaction problem over a state space Ω, with solutions lying in a subset $\Omega_T \subseteq \Omega$. A prompt P induces a partition of Ω into a resolved subspace Ω_P^{given} and an unresolved subspace Ω_P^{active}, such that $\Omega = \Omega_P^{\text{given}} \times \Omega_P^{\text{active}}$ up to equivalence. Execution under prompt P consists in navigating Ω_P^{active} while treating Ω_P^{given} as fixed.

The effective difficulty of T under P is determined not by the size or structure of Ω_T alone, but by the induced geometry of Ω_P^{active}. Two prompts P_1 and P_2 that describe the same task may induce radically different active subspaces, with different dimensionalities, coupling structures, and cost landscapes. A prompt that relegates a highly coupled set of variables into Ω_P^{given} collapses the search space, while a prompt that exposes those same variables renders the task combinatorially explosive.

This formalization makes precise the claim that difficulty is prompt-relative. There exists no canonical prompt-independent representation of a task that preserves difficulty across systems. Any representation privileges some decompositions over others, and those privileges reflect the historical availability of compilations rather than intrinsic task structure.

Prompt relativity also implies that learning can be interpreted as prompt construction. To learn a task is not merely to improve performance within a fixed representation, but to discover prompts that induce tractable active subspaces. Expertise consists in the ability to reframe tasks so that most dependencies are relegated, while failure often reflects being trapped in representations that expose unnecessary structure.

This view explains why minor reframings can produce disproportionate performance gains, and why such gains are often misattributed to increases in intelligence or capacity. What has changed is not the system's power but the geometry of the space in which it is operating.

The relativity of prompts further implies that delegation is a form of prompt selection. When a task is delegated to another agent or tool, a prompt is implicitly chosen that aligns the task with that agent's compiled affordances. Successful delegation minimizes the mismatch between the prompt-induced active subspace and the delegate's existing compilations. Misalignment produces the familiar phenomenon of tasks that are trivial for one agent and baffling for another, even when both are competent within their own domains.

Finally, prompt relativity reinforces the impossibility of stable task classification. Any taxonomy of tasks presupposes a privileged representation in which task identity and difficulty are well defined. But if prompts reshape the active structure of tasks, then task identity itself becomes context-sensitive. What counts as the same task across different prompts is already a negotiated abstraction, not a given.

This appendix therefore formalizes one of the paper's central claims: there is no view from nowhere from which task difficulty can be assessed. Difficulty emerges only after a prompt has been chosen, and the choice of prompt is itself a historical and practical act rather than a neutral description.

G Stability, Recompilation, and Phase Transitions in Difficulty

This appendix formalizes the notion that shifts in difficulty often occur not gradually but through abrupt transitions associated with the breakdown and reconstruction of compiled structure. These transitions resemble phase changes in physical systems, though the relevant variables are cognitive, computational, or institutional rather than thermodynamic.

Let $\mathcal{P}(t)$ denote the set of compiled affordances available to a system at time t, and let $E(t)$ denote the environment in which those affordances are deployed. We say that a compilation $a \in \mathcal{P}(t)$ is stable over an interval $[t_0, t_1]$ if the assumptions embedded in a remain approximately satisfied throughout that interval. Stability here is not binary but graded, depending on the tolerance of a to deviations in environmental parameters.

Difficulty transitions occur when accumulated environmental drift causes one or more critical compilations to cross a stability threshold. Below this threshold, the abstraction continues to function and difficulty remains low. Above it, the abstraction fails catastrophically, forcing the system to re-expose internal dependencies that had previously been relegated. The subjective experience is often one of sudden confusion or overload, while the objective manifestation is a sharp increase in error rates or resource consumption.

This behavior can be modeled by introducing a stability functional $\sigma(a, E)$ that measures the alignment between a compilation and its environment. Difficulty with respect to a task T can then be written schematically as

$$D(T, S, \mathcal{P}, E) \approx \sum_{a \in \mathcal{P}_T} f(a)\, \mathbf{1}_{\sigma(a,E)<\theta},$$

where $\mathcal{P}_T \subseteq \mathcal{P}$ denotes the subset of compilations relevant to T, $f(a)$ measures the contribution of a's failure to total difficulty, and θ is a stability threshold. When $\sigma(a, E) \geq \theta$, the compilation functions and contributes little to difficulty. When $\sigma(a, E) < \theta$, the compilation collapses and its internal complexity is reintroduced.

This formalism captures why difficulty often appears discontinuous. Environmental change may be gradual, but its effect on compiled abstractions is nonlinear. As long as critical assumptions hold, performance remains stable. Once they fail, difficulty spikes. Recompilation may eventually restore ease, but only after significant cost has been incurred to construct new abstractions aligned with the altered environment.

The same logic applies at collective and institutional scales. Bureaucratic procedures, technical standards, and social norms function as large-scale compilations. They dramatically reduce coordination cost under stable conditions but become brittle when underlying realities shift. Institutional crises often correspond to simultaneous failures of multiple compilations whose assumptions have quietly eroded. The resulting surge in difficulty is frequently misdiagnosed as moral failure or incompetence rather than as a structural phase transition.

This perspective also clarifies why systems often oscillate between periods of apparent stability and periods of rapid reorganization. Stability reflects successful compilation under relatively constant conditions. Reorganization reflects forced recompilation triggered by environmental change or internal accumulation of mismatch. There is no final steady state because the act of stabilization itself reshapes the environment, sowing the seeds of its own breakdown.

Understanding difficulty transitions as phase-like phenomena reinforces the paper's central claim. Ease and effort are not smoothly varying properties but emergent states contingent on the viability of abstractions. When abstractions fail, difficulty does not increase incrementally; it returns in bulk. This behavior is not pathological but intrinsic to systems that rely on relegation and compilation to function at all.

The appendix thus provides a formal bridge between the qualitative analysis of noun-free cognition and the observed punctuated dynamics of learning, technological change, and institutional evolution.

H Against Process Reification: Distinction from Process Psychology and E-Prime

The noun-free stance advanced in this paper may appear, at first glance, to align naturally with traditions that emphasize process over substance, such as process psychology or linguistic programs like E-Prime. While there is a superficial resemblance, the alignment is limited. The present framework departs from both in its motivation, scope, and ontological commitments. Clarifying these differences is necessary to avoid misinterpretation.

Process psychology typically replaces static mental objects with dynamic processes, emphasizing flows, transitions, and temporal evolution. While this move is often presented as an alternative to object-based cognition, it frequently preserves the same underlying ontology at a different level of description. Processes are treated as entities with stable identities, governed by laws, stages, or mechanisms that can themselves be cataloged. The nounification is displaced rather than eliminated. What was once a mental object becomes a mental process, but the explanatory structure continues to rely on identifiable units that persist across contexts.

The framework of noun-free cognition rejects this substitution. The goal is not to replace objects with processes, but to deny that either constitutes a privileged explanatory primitive. Difficulty, abstraction, intelligence, and cognition are not redescribed as processes; they are analyzed as relational effects that arise from the interaction of functions operating under constraints. These effects have no stable identity independent of the configurations that produce them. To speak of them as processes risks reintroducing exactly the kind of ontological stability the framework seeks to dissolve.

A similar distinction applies to E-Prime, the linguistic discipline that prohibits the use of forms of the verb “to be” in order to reduce reification and encourage attention to action and relation. E-Prime succeeds as a pedagogical and rhetorical tool, highlighting how easily language smuggles metaphysical commitments into ordinary speech. However, its intervention remains primarily grammatical. It constrains expression without providing an alternative explanatory architecture. One may write in E-Prime while retaining a fundamentally object-oriented model of mind, cognition, or society.

Noun-free cognition does not impose a linguistic prohibition. It permits the use of nouns where they function as convenient compressions, but it insists that such compressions not be mistaken for ontological commitments. The critique operates at the level of explanatory structure rather than syntax. The problem is not the presence of nouns in language, but the assumption that those nouns correspond to stable entities that exist independently of their operational context.

The positive alternative proposed here is neither object-oriented nor process-oriented, but functional and circuit-based. The basic explanatory unit is not a thing or a process, but a transformation under constraint. Systems are characterized by what they do given particular boundary conditions, not by what they are in isolation. Functions compose into circuits, circuits stabilize into scaffolds, and scaffolds temporarily support the relegation of complexity. What appears as an object or a process is a snapshot of a functioning circuit that has become sufficiently stable to be treated as atomic.

This circuit-first ontology aligns naturally with the analysis of abstraction as compilation. A compiled abstraction is not an object stored in the mind, nor a process unfolding over time, but a callable transformation that hides internal structure. When conditions change, the circuit fails, and its internal wiring becomes visible again. Difficulty emerges at that moment, not because a process has slowed or an object has degraded, but because a functional pathway has lost viability.

Distinguishing this view from process psychology is especially important in the context of cognition. Process theories often seek to explain behavior by identifying canonical sequences or stages that recur across tasks. Noun-free cognition denies that such sequences have stable identity outside the contexts that sustain them. What recurs is not a process but a pattern of constraint satisfaction that happens to reappear under similar conditions. Change the conditions, and the pattern dissolves.

The distinction from E-Prime is equally important at the philosophical level. E-Prime aims to discipline description. Noun-free cognition aims to discipline ontology. It does not claim that avoiding reifying language will solve conceptual problems, but that refusing to treat abstractions as entities forces attention onto the dynamics that actually generate behavior. Language may follow from that shift, but it does not substitute for it.

By adopting a function- and circuit-first ontology, the framework preserves explanatory power while avoiding the metaphysical commitments that generate false stability. Objects and processes reappear as useful fictions when conditions permit, but they are never granted foundational status. What remains fundamental are transformations, constraints, and the continual renegotiation of the boundaries that make action tractable at all.

This appendix therefore situates noun-free cognition not as a variant of existing anti-essentialist programs, but as a distinct methodological stance. It refuses both object metaphysics and process metaphysics in favor of an operational view in which meaning, difficulty, and cognition exist only as effects of functioning systems embedded in changing environments.

I Normative and Ethical Consequences of Relational Difficulty

This appendix addresses the normative implications of treating difficulty as relational rather than intrinsic. While the main text remains primarily descriptive, the framework carries ethical consequences that follow directly from its ontology. These consequences concern responsibility, fairness, evaluation, and governance in systems where difficulty is continually displaced rather than eliminated.

If difficulty is produced by the interaction between tasks, prompts, compiled affordances, and environments, then moral judgments that attribute success or failure solely to individual agents are systematically incomplete. Performance cannot be evaluated independently of the scaffolding that made that performance possible. Praise and blame, when attached to outcomes alone, implicitly assume that the difficulty of the task was uniform across agents. Under noun-free cognition, this assumption is false by default. Agents operate within different compilation histories, with unequal access to tools, training, institutional support, and stable environments. What appears as diligence or talent may reflect alignment between prompts and available scaffolds, while what appears as incompetence may reflect a structural mismatch rather than individual deficiency.

This reframing has direct consequences for educational and professional evaluation. Standardized assessments aim to compare agents by holding tasks constant, but they cannot hold prompts, environments, or compiled affordances constant in practice. Even when test items are identical, the effective task differs depending on which abstractions have already been stabilized by prior experience. As a result, assessments systematically conflate aptitude with prior access to scaffolding. Noun-free cognition does not deny the existence of individual differences, but it rejects the inference that outcomes transparently reveal intrinsic capability.

Responsibility must therefore be redistributed across systems rather than localized within agents. When a system imposes prompts that expose dependencies some agents cannot reasonably resolve with their available affordances, the resulting difficulty is system-generated. Ethical responsibility lies not only with the agent who fails, but with the designers of the task environment. This applies equally to workplaces, educational institutions, bureaucracies, and technological interfaces. Difficulty becomes an ethical variable to be managed rather than a natural fact to be endured.

The framework also challenges common narratives about merit and desert. Merit is typically inferred from success under conditions presumed to be fair. If difficulty is mobile and relational, then fairness cannot be defined solely by identical treatment. Fairness requires attention to how difficulty is distributed across agents through differential access to compilation. Equal prompts can generate unequal burdens. Ethical design therefore requires adaptive prompting that aligns demands with available scaffolding or provides mechanisms for recompilation.

At a larger scale, noun-free cognition reframes debates about technological disruption. Automation is often justified on the grounds that it removes difficulty from human labor. The analysis developed here shows that difficulty is instead displaced, frequently onto less visible forms of labor, such as maintenance, monitoring, emotional regulation, and coordination. Ethical evaluation of technology must therefore account for where difficulty migrates and who bears it. A system that simplifies user experience while externalizing complexity onto precarious workers or future generations cannot be ethically assessed by efficiency metrics alone.

This perspective also alters how accountability should be assigned in complex systems. When failures occur, investigations often seek individual fault. Noun-free cognition suggests that many failures reflect misaligned compilations rather than negligence or malice. Accountability, in such cases, should focus on why abstractions were allowed to ossify beyond their validity window and why recompilation mechanisms were absent or suppressed. Ethical responsibility includes maintaining the conditions under which abstractions can be revised without catastrophic cost.

Finally, the framework has implications for how societies reason about progress. If progress is understood as the elimination of difficulty, then recurring crises appear as regressions or failures. If progress is understood as the continual redistribution of difficulty, then the ethical task becomes one of steering that redistribution toward resilience rather than fragility. This involves cultivating systems that can surface hidden complexity before it accumulates destructively and that allow difficulty to be renegotiated rather than denied.

In this sense, noun-free cognition supports a modest but demanding ethical stance. It does not promise a world without difficulty, nor does it excuse harm by appealing to inevitability. Instead, it insists that difficulty is always being produced somewhere, by someone, under some configuration. Ethical responsibility consists in recognizing where it is being produced, who is bearing it, and whether the distribution can be altered through better scaffolding, more honest metrics, or more flexible forms of recompilation.

This appendix completes the transition from descriptive ontology to normative consequence. By refusing to treat difficulty as a thing, it becomes possible to treat it as a shared condition—one that can be managed, redistributed, and, at times, deliberately accepted, but never finally abolished.

J Beyond Use: Extending Anti-Essentialism to the Predictability of Difficulty

The position developed in this paper can be understood as extending, and in a specific sense radicalizing, the anti-essentialist program articulated in the later work of *Wittgenstein*. Wittgenstein's refusal to define language games or family resemblances was grounded in the observation that meaning does not rest on fixed criteria but on patterns of use that shift across contexts. Words function without essence because their application is stabilized only locally and temporarily within practices. Any attempt to capture their meaning in a definition misunderstands the source of their intelligibility.

The present framework accepts this insight but pushes it further. Wittgenstein showed that one cannot define in advance what counts as the correct use of a word across all future contexts. Noun-free cognition adds the claim that one likewise cannot define in advance what counts as an easy or difficult task, even when the task itself appears unchanged. Difficulty, like meaning, is not merely context-sensitive but historically unstable. It can reverse direction, oscillate, or dissolve entirely as the scaffolding that sustains it shifts.

In Wittgenstein's analysis, we know how to go on in a language game without possessing an explicit rule that guarantees correct continuation. The rule-following paradox arises because any rule admits multiple interpretations, and it is practice that resolves the indeterminacy. In the present case, an analogous structure appears at the level of action rather than meaning. We know how to perform tasks without possessing a principle that guarantees their continued tractability. What resolves the indeterminacy is not a rule but a configuration of tools, abstractions, habits, and environmental regularities that happen, for a time, to align.

The crucial extension is this: while Wittgenstein denied the possibility of defining the essence of a practice, he did not explicitly claim that the ease of participation in a practice is itself fundamentally unpredictable. Noun-free cognition makes that stronger claim. Even if a task is perfectly well understood, and even if its criteria of success remain fixed, there is no principled way to predict whether it will remain easy, become difficult, or become easy again. The determinants of difficulty lie not in the task's description but in contingent alignments that cannot be specified exhaustively in advance.

This unpredictability is not merely epistemic, arising from lack of information. It is structural. Any attempt to predict difficulty presupposes that the relevant compilation boundaries will remain intact. Yet those boundaries are altered by the very processes of use, optimization, and abstraction that make tasks tractable in the first place. As a result, predictions about difficulty participate in the dynamics they attempt to forecast. The future ease of a task is entangled with how it is practiced, taught, automated, measured, and valued in the present.

The analogy to language games can now be sharpened. Just as the meaning of a word is renegotiated with each use, the difficulty of a task is renegotiated with each execution. Each performance leaves traces: tools are refined, shortcuts are discovered, assumptions are reinforced, and expectations shift. These traces modify the environment in which the task will next be encountered. Difficulty is therefore not merely context-dependent but reflexive. It depends on its own history.

This reflexivity explains why difficulty can reverse without contradiction. A task may become easy as abstractions accumulate, difficult again as those abstractions misalign with new conditions, and easy once more after recompilation. There is no paradox here because difficulty never resided in the task as such. What changes is the viability of the circuit through which the task is executed. The task's identity remains stable only at the level of description; its operational reality does not.

In this respect, noun-free cognition departs decisively from any view that seeks to ground difficulty in objective task structure. Even a complete specification of a task's rules, inputs, and outputs is insufficient to determine its future tractability. The missing information is not hidden in the task but distributed across evolving systems of practice. To ask whether a task is intrinsically hard is therefore akin to asking whether a word intrinsically means something independent of use. Both questions presuppose a stability that the phenomena themselves do not possess.

The extension beyond Wittgenstein thus lies not in rejecting his insights, but in generalizing them from semantics to action, from meaning to effort, and from language to cognition as a whole. Where Wittgenstein showed that meaning cannot be fixed by definition, this framework shows that difficulty cannot be fixed by analysis. Both are products of living systems that stabilize patterns temporarily and then move on.

Seen this way, the impossibility of predicting task difficulty is not a limitation to be overcome but a consequence of taking anti-essentialism seriously. Once difficulty is stripped of its noun-like status, its mobility becomes inevitable. There is no fact of the matter, independent of history and context, about what will remain easy or hard. There is only the ongoing activity of systems negotiating their own boundaries, one execution at a time.

K A Categorical and Sheaf-Theoretic Formalization of Relational Difficulty

This appendix sketches a category-theoretic and sheaf-theoretic formalization of the central claims of the paper. The objective is not to force a finished axiomatization, but to exhibit a precise mathematical idiom in which the mobility of difficulty, the role of prompts as boundary conditions, and the inevitability of recompilation can be stated without contradiction. The guiding intuition is that tasks, prompts, and affordances should be modeled not as intrinsic objects but as data varying over contexts, with coherence imposed only locally and up to refinement.

K.1 Contexts as a Site and Prompts as Morphisms

Let C be a category of contexts. Objects $U \in \mathrm{Ob}(\mathsf{C})$ represent concrete configurations in which action occurs, including environmental constraints, available tools, institutional conventions, and historical scaffolding as they present themselves at a given scale. A morphism $f : V \to U$ represents a change of context by refinement, restriction, or translation. Refinement should be understood broadly: moving from a coarse description to a finer one, narrowing the operational environment, changing interfaces, or conditioning on additional constraints.

Equip C with a Grothendieck topology J, turning (C, J) into a site. A covering family $\{f_i : U_i \to U\}_{i \in I}$ represents a decomposition of a context U into locally manageable subcontexts U_i whose overlap structure captures the coherence conditions required for global coordination. This expresses, in categorical terms, the idea that tractability is frequently a local phenomenon and that global behavior is assembled from patches.

A prompt is modeled as a morphism that mediates between a task description and the context of execution. Concretely, one may treat prompts as arrows in an auxiliary category P equipped with a functor $\pi : \mathsf{P} \to \mathsf{C}$, where an object of P is a prompt-instance $\langle U, p\rangle$ anchored over $U = \pi(\langle U, p\rangle)$. A change in prompting corresponds to a morphism in P lying over a morphism in C, thereby encoding the principle that prompting is inseparable from context.

K.2 Tasks as Sheaves of Solutions and the Variability of Meaning

Fix a target of evaluation, such as a goal condition, constraint satisfaction predicate, or behavioral specification, abstracted as a task T. The central move is to represent T not as a single set but as a presheaf (and, where appropriate, a sheaf) of realizations over contexts. Define a presheaf

$$\mathcal{S}_T : \mathsf{C}^{\mathrm{op}} \longrightarrow \mathsf{Set},$$

where $\mathcal{S}_T(U)$ is the set of admissible solutions, executions, or successful realizations of task T in context U. For a morphism $f : V \to U$, the restriction map $\mathcal{S}_T(f) : \mathcal{S}_T(U) \to \mathcal{S}_T(V)$ expresses how a realization in the coarser context U induces, by refinement, a realization in V. This captures the operational fact that what counts as "the same task" is enacted through restriction and re-expression across contexts, rather than given by a context-free identity.

When $\mathcal{S}_T$ satisfies the sheaf condition with respect to the topology J, local realizations that agree on overlaps glue uniquely to a global realization. The failure of the sheaf condition encodes precisely the kinds of coordination breakdowns emphasized in the main text: one may have locally valid solutions that cannot be made globally coherent because the overlaps encode constraints that local patches can ignore. In this formal language, the apparent stability of a task is a sheaf-theoretic property rather than an intrinsic one.

This also provides a clean interpretation of Wittgensteinian family resemblance. A "concept" is not a set with a definition but a sheaf-like object whose sections vary with context, with resemblance realized through overlap data rather than a global essence. The refusal to define is then the refusal to pretend that a global section exists where only locally gluable data is available.

K.3 Affordances as a Sheaf, Compilation as a Monoidal Compression

Let $\mathcal{A}$ be a presheaf (often better treated as a sheaf of structures) assigning to each context U the available affordances in that context, such as tools, habits, compiled abstractions, institutional procedures, and callable modules:

$$\mathcal{A} : \mathsf{C}^{\mathrm{op}} \longrightarrow \mathsf{Str},$$

where Str is a category of structured sets, algebras, or computational resources. A refinement $f : V \to U$ induces a restriction $\mathcal{A}(U) \to \mathcal{A}(V)$ representing how affordances degrade, specialize, or become unavailable under tightened constraints, or conversely how additional structure appears when moving to a context that includes more scaffolding.

To model compilation, it is useful to assume Str carries a symmetric monoidal structure $(\otimes, \mathbb{I})$ representing compositional assembly of affordances. A compilation operation is then a family of morphisms in Str that replace a composite with an atomic proxy. One can represent a compilation in context U as a morphism

$$c_U : a_1 \otimes a_2 \otimes \cdots \otimes a_n \longrightarrow \widehat{a}$$

inside $\mathcal{A}(U)$, where $\widehat{a}$ is the compiled abstraction that exposes a stable interface and hides internal structure. The key claim that compilation is provisional is captured by the dependence of c_U on U: there need not exist a natural family $\{c_U\}$ compatible with restriction along morphisms in C. In other words, compilation does not define a natural transformation unless strong stability conditions hold.

This lack of naturality is the categorical expression of “compilation against moving targets.” If $f : V \to U$ is a refinement, the square

$$\begin{tikzcd}
a_1 \otimes \cdots \otimes a_n \arrow[r, "c_U"] \arrow[d] & \widehat{a} \arrow[d] \\
a_1' \otimes \cdots \otimes a_n' \arrow[r, "c_V"] & \widehat{a}'
\end{tikzcd}$$

need not commute. Commutativity would mean that compiling first and then refining yields the same operational artifact as refining first and then compiling. Its failure is precisely the phenomenon that a compiled abstraction may work in U and fail in V, or vice versa, and that recompilation is required after context shift.

K.4 Difficulty as a Costed Functor and the Nonexistence of a Global Ranking

Define a cost object, for example the ordered commutative monoid $(\mathbb{R}_{\geq 0}, +, 0, \leq)$ or a richer resource semiring. Let Cost be a category whose objects are costed structures and whose morphisms preserve the relevant order. A relational notion of difficulty can then be expressed as a functor

$$D_T : \int \mathcal{A} \longrightarrow \mathsf{Cost},$$

from the Grothendieck construction $\int \mathcal{A}$, whose objects are pairs (U, a) with $a \in \mathcal{A}(U)$, to costs. Intuitively, $D_T(U, a)$ is the expected cost of realizing T in context U using affordance configuration a. Morphisms in $\int \mathcal{A}$ represent simultaneous changes of context and affordance under restriction, extension, or translation. The paper's central claim that difficulty is not intrinsic becomes the statement that there is no factorization through T alone; there is no functor $\widetilde{D_T}$ with $D_T(U, a) = \widetilde{D_T}(T)$ independent of (U, a).

One may also express prediction failure as the nonexistence of a global section of a suitable "difficulty classification" sheaf. Consider a presheaf $\mathcal{K}$ that assigns to each context U a classification of tasks into equivalence classes such as "easy" and "hard":

$$\mathcal{K}(U) = \{\text{classifications of } T \text{ in } U\}.$$

If the mobility of difficulty is genuine, then $\mathcal{K}$ will typically fail to have a global section stable under refinement. More strongly, even when local classifications exist on a cover $\{U_i \to U\}$, they need not agree on overlaps $U_i \times_U U_j$, preventing gluing. The obstruction to gluing is the formal expression of the paper's claim that no stable task classification exists across shifting contexts.

K.5 Recompilation as Descent Data and Obstructions as Sheaf Cohomology

Recompilation can be modeled as the process of producing descent data that restores coherence after a context shift. Suppose a context U is covered by $\{U_i \to U\}$ such that tractable execution exists locally on each U_i via compiled affordances $\widehat{a}_i \in \mathcal{A}(U_i)$. To obtain a coherent global affordance on U, one requires compatibility isomorphisms on overlaps

$$\phi_{ij} : \widehat{a}_i|_{U_{ij}} \xrightarrow{\cong} \widehat{a}_j|_{U_{ij}}, \quad U_{ij} = U_i \times_U U_j,$$

satisfying a cocycle condition on triple overlaps. The need for such ϕ_{ij} is exactly the need to coordinate compiled abstractions across subcontexts. Failure to find such descent data corresponds to the familiar phenomenon in which local solutions exist but cannot be integrated into a stable global workflow.

When $\mathcal{A}$ is a sheaf of groups or groupoids (or when one works in a stacky setting), these failures can be organized cohomologically: the obstruction to gluing lives in a nontrivial cohomology class measuring the incompatibility of local compilations. In practice, these obstructions are realized as interface mismatches, institutional frictions, version conflicts, and semantic drift. The sheaf-theoretic point is that such incompatibilities are not accidental; they are the generic case once compilation is context-dependent.

K.6 Goodhart Dynamics as Endomorphisms and Topology Refinement

Goodhart dynamics can also be expressed in this language. A metric is a natural transformation from a sheaf of rich phenomena to a sheaf of measurable quantities. Let $\mathcal{G}$ be a sheaf encoding the underlying goal-related structure of a system, and let $\mathcal{M}$ be a sheaf of measurable proxies. A metric is a morphism of presheaves (ideally of sheaves)

$$m : \mathcal{G} \longrightarrow \mathcal{M}.$$

Optimization with respect to m induces endomorphisms of contexts and affordances, effectively changing the site by privileging certain covers and suppressing others. In operational terms, what becomes measurable becomes selectable, and what becomes selectable becomes a locus of exploitation that alters $\mathcal{G}$ itself. Categorically, the act of optimizing m corresponds to applying an endofunctor $F : \mathsf{C} \to \mathsf{C}$ that reshapes contexts, often forcing refinements in which the original metric no longer behaves as a good proxy. The repeated introduction of new metrics corresponds to iterated topology refinement, in which the covers required for local tractability become increasingly specialized, narrowing the conditions under which global gluing is possible.

This provides a precise sense in which metricization "reifies" a process. It attempts to replace a context-sensitive sheaf $\mathcal{G}$ with a more rigid quantity $\mathcal{M}$, but the induced endomorphisms of the base site destroy the very naturality conditions that made the metric informative. The erosion of predictive validity is then a functorial consequence rather than an empirical surprise.

K.7 Summary of the Formal Picture

Within this categorical and sheaf-theoretic framing, the paper's central claims become structural statements about variance and gluing. Tasks are modeled as sheaves of realizations over a site of contexts; prompts are morphisms that determine which restrictions are operative; affordances form a sheaf of callable structure; compilation is a context-dependent monoidal compression that typically fails to be natural; difficulty is a costed functor on the total category of affordances over contexts; and the impossibility of stable task classification is expressed as the generic nonexistence of global sections for the relevant classification presheaf. Prediction failures and recompilation events correspond to obstructions to descent, often expressible cohomologically in stack-like settings. Goodhart dynamics appear as the iterative deformation of the base site induced by optimizing proxy maps, shrinking the region of contexts where earlier abstractions glued successfully.

This formalization does not remove the mobility of difficulty; it makes that mobility explicit as a mathematical property. The point is not that cognition and society secretly "are" sheaves, but that the relations the paper describes behave like sheaf-valued data: locally coherent, globally fragile, and continually reshaped by the very operations that attempt to stabilize them.

L Corollary: The Asymmetry Between Saying and Doing

This appendix formulates a corollary that follows naturally from noun-free cognition and that supports a strong probabilistic prediction. The corollary states that, under broad and stable assumptions, saying will remain easier than doing. The claim is not that speech acts are effortless, nor that doing is always difficult, but that the act of linguistic or symbolic specification systematically underestimates the cost of execution. This underestimation is structural rather than accidental, and it persists even as tools and environments change.

The basic reason is that saying, writing, or labeling is a form of compression. A description functions as a prompt that partitions the world into resolved and unresolved aspects. In practice, this means that a description can name a generative process without instantiating the full set of affordances and constraints required to realize it. A label can therefore be produced at a cost that is small relative to the cost of building the causal circuit to which the label refers.

This asymmetry can be formalized using a simple generative model. Let X be a target outcome, and let $\mathcal{G}$ be a family of generative procedures that can produce X under some environment E. A statement s is an encoding of X relative to a language $\mathcal{L}$, while an execution $\gamma \in \mathcal{G}$ is a concrete causal pathway that realizes X. Let $C_{\text{say}}(s)$ denote the cost of producing the statement, and let $C_{\text{do}}(\gamma; E)$ denote the cost of executing the generative procedure in environment E. The corollary asserts that, for broad classes of tasks and environments,

$$\mathbb{E}[C_{\text{say}}(s)] \ll \mathbb{E}[C_{\text{do}}(\gamma; E)]\,,$$

where the expectations are taken over plausible tasks, encodings, and environmental conditions.

A more informative inequality compares minimal costs. Let $K_{\mathcal{L}}(X)$ be the description length of X in a language $\mathcal{L}$, and let $A_E(X)$ denote the minimal assembly cost of realizing X in environment E using available primitives and affordances. Then the structural asymmetry can be written as

$$K_{\mathcal{L}}(X) \ll A_E(X)$$

for most nontrivial outcomes X of practical interest. The left-hand side reflects a symbolic compression; the right-hand side reflects the constructive complexity of building a functioning circuit in the world. The two quantities are linked but not comparable by a fixed conversion factor, because $A_E(X)$ depends on fragile, context-specific constraints that do not appear in $K_{\mathcal{L}}(X)$.

This perspective also clarifies why the gap persists under technological progress. Improvements in tools reduce $A_E(X)$ for some outcomes by providing new compiled affordances, but those same improvements also expand the space of outcomes that can be named. As soon as one can do more, one can also say more. Symbolic capacity tends to outpace constructive capacity because naming scales with combinatorial recombination, while doing scales with embodied causality, coordination, and maintenance. The result is that the ratio between saying and doing may fluctuate locally but remains skewed in expectation.

The corollary also explains a familiar social phenomenon: plans proliferate more readily than implementations. A plan is a linguistic object that labels a path through a space of affordances without supplying the affordances themselves. Plans therefore accumulate in domains where execution is bottlenecked by physical constraints, coordination costs, or hidden dependencies. The production of proposals becomes cheap relative to the production of working systems. This is not a moral indictment; it is a consequence of representational compression.

The prediction can be stated in Bayesian terms. Let H be the hypothesis that saying is easier than doing in the sense above, and let E be the evidence supplied by the ubiquity of execution failures, scope creep, underestimated timelines, and the persistent divergence between proposal volume and realized output across domains. Under noun-free cognition, the likelihood $P(E \mid H)$ is high because the hypothesis follows from the structural mismatch between descriptive compression and constructive causality. Competing hypotheses that treat the saying–doing gap as merely contingent or cultural yield lower likelihoods, because they require additional assumptions to explain the cross-domain persistence of the gap. The posterior probability assigned to H is therefore high under broad priors that treat representational compression as cheaper than causal construction.

This corollary does not imply that speech is epistemically inferior or that description is futile. On the contrary, saying is indispensable precisely because it allows coordination around compressed prompts. The corollary instead provides a disciplined warning about what description can and cannot guarantee. A label is not a circuit. To say that a system will exist is to point at a region of possibility space; to build the system is to construct the affordances and constraints that make that region reachable.

Within the framework of noun-free cognition, the saying–doing asymmetry becomes a special case of the general thesis. Difficulty is mobile because compilation is provisional. Saying is easier because it is itself a compiled operation that names what doing must still assemble. This relationship remains stable even as particular tasks move between ease and effort. In that sense, the corollary provides one of the few robust directional predictions available in a world where intrinsic task difficulty cannot be forecast.

M On the Redundancy of Further Formalization

It may appear that the preceding appendices invite yet another level of abstraction in order to complete the argument. This appendix exists only to deny that necessity. The central claims of the paper do not depend on any particular formalism, nor do they require a final theoretical ladder to be constructed and defended. On the contrary, the framework of noun-free cognition already implies that any such ladder, once used, must be dismantled.

The notion of a scaffold captures this implication directly. A scaffold is erected to enable construction under conditions where direct access is impossible or unsafe. Once the structure it supports is complete, the scaffold becomes not merely unnecessary but obstructive. Its continued presence interferes with the very function it once enabled. In the same way, conceptual, mathematical, and linguistic scaffolds serve their purpose only insofar as they remain provisional. Their value lies in what they make possible, not in their preservation.

All of the formal tools introduced in this paper—difficulty functions, compilation sketches, assembly indices, categorical diagrams, and sheaf-theoretic constructions—are scaffolds in precisely this sense. They are aids to orientation, not ontological commitments. They allow certain relations to be seen clearly, at the cost of temporarily reifying what the argument insists must remain fluid. Once those relations are grasped, the scaffolding can and should be removed.

This conclusion is not an expression of anti-formalism. It is a recognition of the role that formalisms play within adaptive systems. Formalizations are themselves compiled abstractions. They stabilize meaning locally, compress reasoning, and facilitate coordination among those who share them. But they also embed assumptions about context, scale, and use. To treat them as final is to mistake their utility for truth.

The ladder metaphor, often invoked in discussions of conceptual clarification, is therefore not an external analogy but an internal consequence of the framework. Any ladder that helps one see why difficulty is mobile will, by that very success, cease to be necessary. To insist on its retention would contradict the insight it provided.

This appendix thus closes the formal sequence by making explicit what has already been implicit throughout. The argument does not culminate in a new object, theory, or definition. It culminates in a way of proceeding: one that remains alert to the provisional nature of its own tools, and that treats understanding not as the accumulation of structures, but as the timely dismantling of those that have done their work.

In that sense, no further appendix is required. The absence of a final foundation is not a deficiency of the framework, but its most faithful expression.

References

[1] L. Wittgenstein. *Philosophical Investigations.* Translated by G. E. M. Anscombe. Blackwell, Oxford, 1953.

[2] D. Kahneman. *Thinking, Fast and Slow.* Farrar, Straus and Giroux, New York, 2011.

[3] H. A. Simon. *The Sciences of the Artificial.* Third edition. MIT Press, Cambridge, MA, 1996.

[4] D. R. Hofstadter. *Gödel, Escher, Bach: An Eternal Golden Braid.* Basic Books, New York, 1979.

[5] G. J. Chaitin. *Meta Math!: The Quest for Omega.* Pantheon Books, New York, 2005.

[6] M. Mitchell. *Complexity: A Guided Tour.* Oxford University Press, Oxford, 2009.

[7] S. Wolfram. *A New Kind of Science.* Wolfram Media, Champaign, IL, 2002.

[8] L. Cronin, S. Walker, B. K. Nicholson, and G. R. J. Cooper. Assembly theory and the selection of molecules. *Journal of the Royal Society Interface*, 17(169), 2020.

[9] C. A. E. Goodhart. Problems of monetary management: The U.K. experience. In *Papers in Monetary Economics*, Reserve Bank of Australia, Sydney, 1984.

[10] M. Strathern. Improving ratings: Audit in the British university system. *European Review*, 5(3):305–321, 1997.

[11] J. C. Scott. *Seeing Like a State: How Certain Schemes to Improve the Human Condition Have Failed.* Yale University Press, New Haven, 1998.

[12] D. H. Autor. Why are there still so many jobs? The history and future of workplace automation. *Journal of Economic Perspectives*, 29(3):3–30, 2015.

[13] D. F. Noble. *Forces of Production: A Social History of Industrial Automation.* Oxford University Press, New York, 1984.

[14] S. Zuboff. *In the Age of the Smart Machine.* Basic Books, New York, 1988.

[15] A. Clark. *Supersizing the Mind: Embodiment, Action, and Cognitive Extension.* Oxford University Press, Oxford, 2008.

[16] G. Marcus and E. Davis. *Rebooting AI: Building Artificial Intelligence We Can Trust.* Pantheon Books, New York, 2019.

[17] H. Moravec. *Mind Children: The Future of Robot and Human Intelligence.* Harvard University Press, Cambridge, MA, 1988.

[18] S. A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution.* Oxford University Press, Oxford, 1993.

[19] K. Friston. The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, 11(2):127–138, 2010.

[20] J. L. England. *Every Life Is on Fire: How Thermodynamics Explains the Origins of Living Things.* Basic Books, New York, 2020.

[21] F. W. Lawvere. Adjointness in foundations. *Dialectica*, 23(3–4):281–296, 1969.

[22] S. Mac Lane. *Categories for the Working Mathematician.* Second edition. Springer, New York, 1998.

[23] A. Grothendieck. Revêtements étales et groupe fondamental. In *SGA 1.* Springer Lecture Notes in Mathematics, Vol. 224, 1971.

[24] T. Pantev, B. Toën, M. Vaquié, and G. Vezzosi. Shifted symplectic structures. *Publications Mathématiques de l'IHÉS*, 117(1):271–328, 2013.