Intelligence as Organized Difficulty Compression

Representation, emergence, and theorem intelligence

Why smarter systems solve harder problems by finding better problem spaces

Dr. Tamas Nagy
`tnagyphd@gmail.com`

Draft • March 2026

Executive Summary (Non-Technical)

The word **intelligence** is overloaded. It is used for benchmark performance, adaptability, reasoning, fluency, planning, and sometimes even consciousness. This paper takes a narrower question: **what is intelligence at the level of a working system?**

The central answer is sharper than the earlier draft of this line. Intelligence is not best understood as raw success inside a fixed problem space. A major source of intelligence is the ability to **reduce the effective difficulty** of a task by finding:

- better representations,
- better decompositions,
- better intermediate objects,
- better state descriptions,
- and better reusable structures.

That is why the paper's core slogan is:

> intelligence is organized difficulty compression.

The paper still keeps the earlier system machinery. Intelligence remains objective-relative, budgeted, and layered. The five recurring layers are perception, modeling, decision, learning, and transfer. A system is not called intelligent merely because it performs well once. It becomes intelligent in the strong system sense when these layers form a stable adaptive organization rather than isolated local skills.

But the deeper claim of the paper is new. Smarter systems often do not win by searching the same space harder. They win by **finding a better space in which to search**.

This is especially visible in theorem proving and research. A difficult theorem is often not made feasible mainly by more brute-force local steps. It becomes feasible after the discovery of:

- the right lemma,
- the right invariant,
- the right decomposition,
- the right bridge object,
- or the right reformulation of the goal.

For that reason, theorem intelligence is treated here as a privileged case of a more general phenomenon: **intermediate object discovery**.

The same idea also explains why some collective systems become smarter while others merely become busier. A collective becomes genuinely more intelligent when it externalizes state, verification, negative knowledge, and representation search into a stable shared organization. In that case the system lowers future difficulty rather than merely increasing the number of parallel guesses.

This main paper has been deliberately tightened around that spine. Historical survey, broader operationalization notes, and implementation-oriented companion material from the wider draft are preserved in aux.md.

Abstract

We propose a system-level theory of intelligence centered on a single claim: **intelligence is organized difficulty compression**. The motivating problem is that the term *intelligence* is routinely asked to cover too much at once, ranging from consciousness and general reasoning to benchmark success and adaptive control. This paper isolates the system-design layer. Let an adaptive system interact with an environment through a history

$$h_t = (o_0, a_0, o_1, a_1, \dots, o_t),$$

with internal state z_t, objective functional J, and resource budget B. We retain a layered, objective-relative description of system intelligence:

$$\mathbf{I}(\pi; \mathcal{E}, J, B) = (P, M, D, L, T),$$

where P is perception of objective-relevant state, M is modeling adequacy, D is decision quality, L is learning gain, and T is transfer to adjacent task families.

The paper's main advance is to argue that a major source of intelligence lies upstream of local action quality. Intelligent systems do not only search harder. They often solve harder problems by discovering better representations ρ that lower the task's effective requirement profile

$$\mathbf{K}_{\text{eff}}(\tau; \rho, B).$$

This yields a difficulty-frontier view of intelligence magnitude and leads to a stronger thesis: frontier lift can arise from representational improvement even when raw solver strength is unchanged.

We also retain an emergence view of intelligence. A merely reactive system is not yet intelligent in the strong system sense used here. Intelligence appears when state gain, model gain, decision gain, learning gain, and transfer gain jointly cross a viability threshold. From this framework we derive three central consequences: reactive policies are insufficient for strong intelligence emergence; the emergence threshold is bottleneck-governed rather than fully compensatory; and for many task families, especially theorem search, representation gain can dominate equal-cost increases in local search throughput.

The theorem domain is treated as a privileged case study. There, intelligence is often expressed not mainly as brute proof search but as **intermediate object discovery**: the invention of lemmas, invariants, decompositions, normal forms, and bridge objects that reduce effective difficulty for the remaining proof obligations. Collective intelligence is then reinterpreted as **externalized cognitive closure**: a shared organization that preserves state, verification, negative knowledge, and reusable representations over time.

The paper is conceptual rather than empirical, but it is not merely verbal. Its goal is to give a sharper object language for distinguishing raw search from representational intelligence, and local competence from systems that genuinely compress difficulty under bounded cost.

---

1. Introduction

1.1 The Real Question

Few words in the study of mind and machines are asked to do more work than *intelligence*. In one context it means reasoning power. In another it means adaptive success. In another it means benchmark rank, planning, verbal fluency, or something close to consciousness. The result is a term that often generates more heat than structure.

For engineering and research-system design, this is not good enough. If one wants to build a system that discovers theorems, routes work well, avoids stale directions, learns from failure, and reuses what it has learned, the question cannot remain a philosophical slogan. It has to become a systems question.

The question pursued here is therefore narrower:

> what makes one bounded adaptive system able to solve problems that another system cannot solve under the same budget?

1.2 Central Claim

The central claim of this paper is:

> At system level, intelligence is organized difficulty compression: the capacity to improve the future state of action by discovering and exploiting structure that lowers the effective difficulty of a task under uncertainty and bounded cost.

This claim has four immediate implications.

First, intelligence is **objective-relative**. There is no useful objective-free scalar that ranks all systems for all purposes.

Second, intelligence is **layered**. Different systems may be strong in decision but weak in learning, or strong in modeling but weak in transfer.

Third, intelligence is **emergent**. A system is not intelligent in the strong organizational sense merely because it performs well once. Intelligence appears when several adaptive layers become jointly viable.

Fourth, intelligence is often **representational**. A major source of system improvement comes not from searching the same problem harder, but from finding a better problem formulation.

1.3 Contributions

The paper makes five contributions.

1. It retains a layered capability object $\mathbf{I} = (P, M, D, L, T)$ for objective-relative system intelligence.

2. It defines an emergence threshold using state, model, decision, learning, and transfer gains.
3. It introduces an explicit distinction between raw task difficulty and **effective** difficulty under a chosen representation.
4. It argues that representation creation is upstream of many practical intelligence gains.
5. It identifies theorem intelligence as a privileged case of **intermediate object discovery**.

1.4 Scope

This paper does **not** try to settle the metaphysics of mind, consciousness, or a universal human-machine ranking. It is a systems paper. Its target is the level at which one designs, evaluates, and compares bounded adaptive organizations.

The broader historical survey, operationalization sketch, and implementation-oriented materials from the previous wider draft are preserved in aux.md so that the main argument can remain sharp.

---

2. Formal Setup

2.1 Objective-Relative Dynamics

We model an adaptive system interacting with an environment $\mathcal{E}$ through

$$\begin{aligned} s_{t+1} &= F(s_t, a_t, \xi_t), \\ o_t &= \Omega(s_t, \eta_t), \\ z_{t+1} &= U(z_t, o_t, a_t), \\ a_t &= \pi(z_t), \end{aligned}$$

where:

- s_t is latent environment state,
- o_t is observation,
- z_t is internal state,
- a_t is action,
- ξ_t, η_t capture exogenous and observational noise.

An objective functional J scores trajectories, and execution is constrained by a budget B.

Not all latent state matters equally. What matters is the **objective-relevant state**

$$r_t = R_J(s_t),$$

the portion of the environment that actually matters for improving J.

2.2 Intelligence As A Capability Vector

The first formal object of the paper is

$$\mathbf{I}(\pi; \mathcal{E}, J, B) = (P, M, D, L, T),$$

where:

- P is perception of objective-relevant state,
- M is modeling adequacy,
- D is decision quality,
- L is learning gain from feedback,
- T is transfer to adjacent task families.

The point of the vector view is diagnostic rather than decorative. A scalar score hides why one system wins and where another fails.

2.3 The Layer Definitions

The coordinates can be read abstractly as follows.

$$P = \mathbb{E}[\mathcal{U}(r_t \mid h_{t-1}) - \mathcal{U}(r_t \mid h_t)],$$

so that perception measures reduction of uncertainty about the state that matters.

$$D = \mathbb{E}\left[\frac{J_{t+1} - J_t}{c_t + \varepsilon}\right],$$

so that decision quality measures objective improvement per unit cost.

$$L = \mathbb{E}[\Phi(\mathbf{I}_{t+1}) - \Phi(\mathbf{I}_t)],$$

for a monotone summary Φ, so that learning measures future capability improvement rather than merely current output.

Modeling M and transfer T are not reducible to one canonical formula across all domains. What matters is their role: modeling captures action-relevant internal structure, while transfer captures reuse of useful structure on adjacent task families.

2.4 Scalar Scores Are Projections

Any scalar score is downstream:

$$I_w = w \cdot \mathbf{I}.$$

This is useful, but only after the objective and evaluator have selected a weighting.

Proposition 1 (No objective-free decision-sufficient scalar). If two systems exchange rank under two legitimate weightings w and w', then no fixed scalar ordering can be decision-sufficient for both objectives.

Proof. If $I_w(A) > I_w(B)$ but $I_{w'}(A) < I_{w'}(B)$, then any fixed total ordering agrees with at most one objective. ☐

This proposition matters because it blocks the temptation to confuse one benchmark or one scalar with the essence of intelligence.

---

3. Emergence As Organized Adaptive Closure

3.1 Reactive And Intelligent Organizations

Consider the reactive class

$$\Pi_{\text{react}} = \{\phi : a_t = \phi(o_t)\},$$

and the model-bearing adaptive class

$$\Pi_{\text{int}} = \{\pi : z_{t+1} = U(z_t, o_t, a_t),\ a_t = \pi(z_t)\}.$$

The qualitative jump is

$$\text{reaction} \longrightarrow \text{model-bearing adaptive closure}.$$

The paper's claim is that intelligence in the strong system sense begins at that organizational jump, not at arbitrary levels of local performance.

3.2 Emergence Gains

Let $r_t = R_J(s_t)$. Define:

$$\begin{gathered}
G_P = I(r_t; z_t) - I(r_t; o_t), \\
G_M = \operatorname{corr}(\widehat{\Delta J_t}(a), \Delta J_t(a)), \\
G_D = J(\pi) - \sup_{\phi \in \Pi_{\text{react}}} J(\phi), \\
G_L = J(\pi^+) - J(\pi), \\
G_T = \mathbb{E}_{\tau' \in \mathcal{N}(\tau)}\left[J_{\tau'}(\pi_{\tau \to \tau'}) - J_{\tau'}(\pi_{\text{scratch}})\right].
\end{gathered}$$

These define the emergence indicator

$$\chi_{\text{int}}(\pi) = \mathbf{1}\{G_P \geq \theta_P,\ G_M \geq \theta_M,\ G_D \geq \theta_D,\ G_L \geq \theta_L,\ G_T \geq \theta_T\},$$

and the bottleneck order parameter

$$\Omega_{\text{int}}(\pi) = \min\left\{\frac{G_P}{\theta_P}, \frac{G_M}{\theta_M}, \frac{G_D}{\theta_D}, \frac{G_L}{\theta_L}, \frac{G_T}{\theta_T}\right\}.$$

3.3 Two Structural Consequences

Proposition 2 (Reactive insufficiency). If $\pi \in \Pi_{\text{react}}$ and carries no additional organization beyond the present observation, then $G_P \leq 0$, $G_L = 0$, and $G_T = 0$. For strictly positive thresholds, $\chi_{\text{int}}(\pi) = 0$.

Proof sketch. A purely reactive policy carries no extra objective-relevant state beyond the current observation, stores no reusable structured update under bounded feedback, and preserves no transferable organization across adjacent tasks. Hence the threshold fails in multiple coordinates. □

Proposition 3 (Non-compensation at the emergence threshold). If $\Omega_{\text{int}}(\pi) < 1$, the emergence threshold is not crossed, regardless of how large the remaining normalized gains are.

Proof. Immediate from the minimum definition of Ω_{int}. □

These propositions formalize two intuitions: intelligence emergence is not equivalent to local success, and its threshold is bottleneck-governed.

---

4. Intelligence As Effective Difficulty Compression

4.1 Difficulty Profiles

Let τ be a task family under budget B. Write its requirement profile as

$$\mathbf{K}(\tau; B) = (\kappa_P, \kappa_M, \kappa_D, \kappa_L, \kappa_T),$$

where the coordinates measure how much diagnostic burden, model burden, decision burden, update burden, and transfer burden the task imposes.

This is not a claim that all tasks come with one canonical natural coordinate system. It is a way of making task difficulty explicit in the same language used to describe intelligence.

We write $x \succeq y$ for coordinatewise dominance and $x \succ y$ when the dominance is strict in at least one coordinate. For a fixed representation ρ, it is useful to name the effective frontier explicitly:

$$\mathcal{F}_{\text{eff}}(\pi; B, \rho) = \{\tau : \mathbf{I}(\pi; \mathcal{E}, J, B) \succeq \mathbf{K}_{\text{eff}}(\tau; \rho, B)\}.$$

4.2 Assumption Schema For The Compression Theorems

The propositions in this section do not claim to be representation-free truths about all possible systems. They are conditional statements inside a specific formal model. The standing assumptions are:

A1 (Task identity under representation). Two representations ρ and ρ' of a task family τ preserve the same underlying task identity: they change the effective burden profile, not the target objective itself.

A2 (Comparable burden coordinates). The coordinates of $\mathbf{K}_{\text{eff}}(\tau; \rho, B)$ live in the same ordered space as the corresponding coordinates of $\mathbf{I}(\pi; \mathcal{E}, J, B)$, so coordinatewise comparisons are meaningful.

A3 (Budget matching). When two representations or two solver variants are compared, the budget B is held fixed unless stated otherwise.

A4 (Local-throughput perturbation model). When a theorem below speaks of increased local search throughput under a fixed representation, it refers to an equal-cost perturbation family that changes local proposal density or local action enumeration while leaving the represented task and its effective burden profile fixed.

A5 (Representation admissibility). When a representation change is treated as legitimate intelligence gain, it is assumed not to change the truth conditions of the target task, only the effective route by which the task is addressed.

These assumptions are mild enough for the current systems-theoretic purpose, but strong enough to prevent slippage between "changing the problem" and "changing the representation of the same problem."

4.3 Robust Solvability And The Frontier

The practical rule is:

> a system solves a task family robustly only if its effective layered capability dominates the task's layered requirement profile with enough margin under the given budget.

Formally, a robust solve requires

$$\mathbf{I}(\pi; \mathcal{E}, J, B) \succeq \mathbf{K}(\tau; B),$$

together with $\Omega_{\text{int}}(\pi) \geq 1$ when the strong organizational notion of intelligence is intended.

This leads to the difficulty frontier

$$\mathcal{F}(\pi; B) = \{\tau : \mathbf{I}(\pi; \mathcal{E}, J, B) \succeq \mathbf{K}(\tau; B)\}.$$

Intelligence magnitude is better understood through the size, shape, and depth of $\mathcal{F}(\pi; B)$ than through one benchmark number.

4.4 Effective Difficulty

Now comes the main move. Let ρ denote a representation, decomposition, abstraction, state choice, or object-language choice for the same underlying task family. The system does not interact only with a raw task. It interacts with a represented task

$$\rho(\tau),$$

whose effective burden we write as

$$\mathbf{K}_{\text{eff}}(\tau;\rho,B).$$

Different ρ-choices can make the same underlying task much easier or much harder for a fixed system. This is where the central thesis enters:

> intelligence often advances by lowering $\mathbf{K}_{\text{eff}}$, not only by increasing raw solver strength.

4.5 Frontier Lift By Representation Change

Proposition 4 (Effective-difficulty reduction expands the frontier). Under Assumptions A1-A3, suppose two representations ρ and ρ' of the same task family satisfy

$$\mathbf{K}_{\text{eff}}(\tau;\rho',B) \preceq \mathbf{K}_{\text{eff}}(\tau;\rho,B),$$

with strict improvement in at least one coordinate. Then any fixed system whose capability vector dominates $\mathbf{K}_{\text{eff}}(\tau;\rho',B)$ but not $\mathbf{K}_{\text{eff}}(\tau;\rho,B)$ can solve the task under ρ' but not under ρ. Hence frontier lift can arise from representational improvement even when raw capability is unchanged.

Proof. By assumption,

$$\mathbf{I}(\pi;\mathcal{E},J,B) \succeq \mathbf{K}_{\text{eff}}(\tau;\rho',B),$$

so by definition $\tau \in \mathcal{F}_{\text{eff}}(\pi;B,\rho')$. Also by assumption,

$$\mathbf{I}(\pi;\mathcal{E},J,B) \not\succeq \mathbf{K}_{\text{eff}}(\tau;\rho,B),$$

so $\tau \notin \mathcal{F}_{\text{eff}}(\pi;B,\rho)$. Thus the same fixed system solves the task under ρ' but not under ρ, and the frontier expands by representational improvement alone. □

This proposition is the paper's first flagship statement. It makes precise the idea that one can become more intelligent by finding a better problem space.

---

5. Representation Creation Is Upstream Of Search

5.1 Upstream Intelligence

Many practical intelligence gains are upstream of the final local action step. Examples include:

- discovering a latent state variable that makes the problem observable,
- discovering a decomposition that turns one large search into several smaller ones,
- discovering an invariant that collapses many cases into one,
- discovering a useful intermediate object such as a lemma, abstraction, or summary statistic,
- discovering a blocker taxonomy that prevents repeating the same false route.

These are not merely cosmetic reframings. They change the geometry of the task.

5.2 Representation Gain Versus Local Throughput

Proposition 5 (Representation gain can dominate bounded local search gain). Under Assumptions A1-A4, There exist systems, task families, budgets, and representations ρ, ρ' such that a representation change yields frontier lift while any bounded equal-cost increase in local search throughput under the worse representation does not.

Proof. Consider a system with capability vector

$$\mathbf{I} = (1, 1, 1, 1, 1),$$

and let $e_D = (0, 0, 1, 0, 0)$. Model bounded equal-cost local-throughput improvement under a fixed representation ρ by the family

$$\mathbf{I}_\alpha = \mathbf{I} + \alpha e_D, \qquad \alpha \in [0, \bar{\alpha}].$$

Now choose a task family τ with effective burden

$$\mathbf{K}_{\text{eff}}(\tau; \rho, B) = (1, 1 + \gamma, 1, 1, 1),$$

for some $\gamma > 0$, and choose $\bar{\alpha} > 0$ arbitrarily. Under ρ, the only violating coordinate is M, so for every $\alpha \in [0, \bar{\alpha}]$,

$$\mathbf{I}_\alpha \not\succeq \mathbf{K}_{\text{eff}}(\tau; \rho, B),$$

because the M-coordinate remains $1 < 1 + \gamma$. Hence

$$\tau \notin \mathcal{F}_{\text{eff}}(\pi_\alpha; B, \rho) \qquad \text{for all } \alpha \in [0, \bar{\alpha}].$$

Now choose a better representation ρ' with

$$\mathbf{K}_{\text{eff}}(\tau; \rho', B) = (1, 1, 1, 1, 1).$$

Then

$$\mathbf{I} \succeq \mathbf{K}_{\text{eff}}(\tau; \rho', B),$$

so $\tau \in \mathcal{F}_{\text{eff}}(\pi; B, \rho')$. Thus the representation change yields frontier lift, while every bounded equal-cost throughput increase under ρ fails to do so. □

This is why notation, decomposition, abstraction, route choice, and state choice are intelligence-bearing moves rather than cosmetic preferences.

5.3 Bottleneck Relief As A Formal Compression Mechanism

The previous propositions can be strengthened in a bottleneck form. Because robust solvability is coordinatewise and the emergence condition is bottleneck-governed, a representation is especially valuable when it reduces the dominant burden coordinate that currently blocks the task.

Proposition 6 (Bottleneck relief by representation change). Under Assumptions A1-A3, Let τ be a task family and suppose there is a coordinate $j \in \{P, M, D, L, T\}$ such that under a representation ρ,

$$I_j(\pi; \mathcal{E}, J, B) < K_{\text{eff},j}(\tau; \rho, B),$$

while for every other coordinate $i \neq j$,

$$I_i(\pi; \mathcal{E}, J, B) \geq K_{\text{eff},i}(\tau; \rho, B).$$

If a new representation ρ' leaves all non-bottleneck effective burdens unchanged or lower and strictly lowers the bottleneck burden so that

$$I_j(\pi; \mathcal{E}, J, B) \geq K_{\mathrm{eff},j}(\tau; \rho', B),$$

then τ moves from outside the robust frontier to inside it without any increase in raw capability.

Proof. Under ρ, the task is outside the frontier because at least one coordinate requirement exceeds the corresponding capability. Under ρ', every coordinate requirement is weakly lower and the unique violating coordinate is brought below capability. Therefore the coordinatewise robust-solve condition now holds, so $\tau \in \mathcal{F}_{\mathrm{eff}}(\pi; B, \rho')$. $\square$

This proposition makes explicit why many important intelligence gains look qualitatively discontinuous. The system may appear stuck until one representational bottleneck is relieved, after which the task becomes solvable without any increase in generic local strength.

5.4 What This Formal Core Does And Does Not Prove

The compression theorems above prove conditional statements inside a representation-sensitive solvability model. They do **not** prove that every real intelligence gain must take this form, nor that all representation changes are beneficial, nor that the five-layer decomposition is the only valid one.

What they do prove is narrower and still important:

1. representational improvement can expand a task frontier without raw capability growth;
2. bounded local search gain can fail where representational change succeeds;
3. relieving the active bottleneck coordinate is enough to move a task into the effective frontier.

These are model-internal theorems, not metaphysical definitions of intelligence in the strongest possible sense.

5.5 A Minimal Proof-Assistant Core

The cleanest next formalization step is to separate the algebraic core from the intelligence-specific interpretation. Fix a finite coordinate type $\mathcal{C}$, an ordered codomain R, a capability profile $I : \mathcal{C} \to R$, and an effective burden profile $K : \mathcal{C} \to R$. Define

$$\mathrm{Solve}(I, K) \iff \forall c \in \mathcal{C},\ K(c) \leq I(c).$$

Then the central formal statements of this paper reduce to a small order-theoretic spine:

1. **monotone frontier inclusion**: if $K' \preceq K$ and $\mathrm{Solve}(I, K')$, then K' is no harder than K for the same I;
2. **strict frontier separation**: if $\mathrm{Solve}(I, K')$ but not $\mathrm{Solve}(I, K)$, then a representational change has created frontier lift for that fixed I;
3. **bottleneck crossing**: if exactly one coordinate blocks solvability under K, and a new burden profile K' lowers that coordinate below capability without worsening the others, then $\mathrm{Solve}(I, K')$.

This matters because Propositions 4 and 6 are already almost entirely of this form. Proposition 5 is then a small constructive separation example inside the same finite-coordinate model. In other words, the first proof-assistant target is not a theory of intelligence in full semantic richness. It is a finite order-theoretic kernel plus a few concrete witnesses.

For later mechanization, the real modeling difficulty therefore lies less in the proofs themselves than in the choice of faithful encodings for:

- what counts as a representation change,
- what counts as admissible burden reduction,
- and how theorem-side obligation structure induces an effective burden profile.

---

6. Theorem Intelligence As Intermediate Object Discovery

6.1 Why Theorem Search Is A Privileged Case

The theorem domain makes the previous claim especially clear. A difficult proof is often not made feasible primarily by trying more local rewrites. It becomes feasible after the discovery of one or more of the following:

- the right lemma,
- the right invariant,
- the right normal form,
- the right bridge object,
- the right case split,
- or the right reformulation under which the statement becomes structurally simpler.

In other words, theorem intelligence is centrally about **intermediate object discovery**.

6.2 A Formal Reading In Terms Of Obligation Sets

To tighten the theorem-side reading, let a represented theorem task (τ, ρ) induce a finite obligation multiset

$$\mathcal{O}(\tau, \rho) = \{o_1, \dots, o_m\},$$

equipped with a nonnegative obligation-complexity functional

$$C(\mathcal{O}) = \sum_{o \in \mathcal{O}} c(o).$$

Think of $c(o)$ as a local proof burden and $C(\mathcal{O})$ as the aggregate remaining burden.

Call an intermediate object q **admissible** for (τ, ρ) if adjoining q yields a new representation $\rho \oplus q$ such that:

1. theorem truth is preserved,
2. the induced obligation set is transformed to $\mathcal{O}(\tau, \rho \oplus q)$,
3. and

$$C(\mathcal{O}(\tau, \rho \oplus q)) < C(\mathcal{O}(\tau, \rho)).$$

This formalizes the intuitive idea that a good lemma, invariant, or bridge object is valuable because it changes the represented proof obligations rather than merely adding one more local step.

We also make one theorem-side monotonicity assumption explicit.

A6 (Theorem-side burden monotonicity). At least one active coordinate of $\mathbf{K}_{\text{eff}}(\tau; \rho, B)$ is monotone in the aggregate obligation complexity $C(\mathcal{O}(\tau, \rho))$ over the representation family being compared.

Proposition 7 (Intermediate-object discovery as theorem compression). Under Assumptions A1-A5 and A6, If an admissible intermediate object q produces a representation $\rho' = \rho \oplus q$ that lowers the active bottleneck burden below capability, then the theorem task enters the effective frontier under ρ' without any increase in raw capability.

Proof. By admissibility, ρ' preserves theorem truth while lowering obligation complexity. By the theorem-side burden monotonicity, this lowers at least the active bottleneck coordinate of $\mathbf{K}_{\text{eff}}(\tau; \rho, B)$. If the lowered coordinate crosses below system capability while the remaining coordinates are unchanged or lower, Proposition 6 applies and the theorem task enters $\mathscr{F}_{\text{eff}}(\pi; B, \rho')$. □

6.3 Corollary For Theorem Systems

Corollary 8 (Theorem frontier lift is often object-mediated). For theorem-search systems, a substantial part of intelligence magnitude is explained not by brute local proof power alone but by the ability to discover intermediate mathematical objects that reduce effective difficulty for the remaining proof obligations.

This corollary is now the joint theorem-side reading of Propositions 4, 6, and 7. A lemma is not merely a step on the path. It is often a representational transformation of the path.

6.4 A Minimal Formalization Interface For Theorem Tasks

If one wants to continue from the order-theoretic core toward a proof-assistant treatment of theorem intelligence, the next layer can be isolated as a narrow interface rather than a full cognitive model. The needed ingredients are:

1. a represented theorem task (τ, ρ);
2. a finite obligation container $\mathcal{O}(\tau, \rho)$;
3. a nonnegative aggregate complexity functional C;
4. an admissibility predicate saying that q preserves theorem truth;
5. a designated active burden coordinate j;
6. a monotone burden-realization map
$$b_j : \mathbb{N} \to \mathbb{N}$$
   such that on the representation family under study,
$$K_{\text{eff},j}(\tau; \rho, B) = b_j(C(\mathcal{O}(\tau, \rho))).$$

Under this narrower interface, admissible intermediate-object discovery factors into two clean steps:

1. **complexity transport**: if q is admissible, then
$$C(\mathcal{O}(\tau, \rho \oplus q)) \le C(\mathcal{O}(\tau, \rho));$$
2. **burden transport**: by monotonicity of b_j,
$$K_{\text{eff},j}(\tau; \rho \oplus q, B) \le K_{\text{eff},j}(\tau; \rho, B).$$

Under that interface, Proposition 7 is no longer mysterious. It becomes a transport theorem from verified obligation-complexity reduction to burden reduction, and then from burden reduction to frontier entry via Proposition 6.

This is the right proof-program boundary for future mechanization. The order-theoretic lemmas should be proved first. The theorem-domain layer should then be added only through this minimal interface, rather than through an overly ambitious global formalization of “research intelligence.”

The key point is that the theorem-side mechanization does not need a full formal theory of creativity or research strategy. It only needs a verified transport interface from obligation simplification to burden relief.

The simplest first instantiation is already enough to be useful. One may take the obligation container to be a finite list of nonnegative local burdens,

$$\mathcal{O}(\tau, \rho) = [n_1, \dots, n_m],$$

with aggregate complexity

$$C(\mathcal{O}) = \sum_{k=1}^{m} n_k.$$

If a verified intermediate object refines the represented proof state so that the new obligation list is pointwise no harder than the old one, then the aggregate complexity weakly decreases automatically. In that setting, theorem-side transport reduces to three small lemmas:

1. pointwise obligation refinement implies aggregate-complexity reduction;
2. aggregate-complexity reduction implies active-burden reduction through b_j;
3. active-burden reduction plus non-worsening of the other coordinates implies frontier entry.

This is mathematically modest, but strategically important: it gives a first concrete theorem-object model in which the paper’s core claim is already mechanizable.

A slightly richer next instantiation is also still manageable. Instead of using bare local burdens, attach a small obligation type label to each entry, for example:

- local rewrite obligation,
- bridge obligation,
- invariant-establishment obligation,
- blocker-resolution obligation.

Then define a weighted aggregate complexity

$$C_w(\mathcal{O}) = \sum_{o \in \mathcal{O}} w(\mathrm{kind}(o)) + \mathrm{burden}(o),$$

where blocker-style obligations may carry larger fixed weight because they are more structurally harmful to progress. Under same-kind pointwise refinement, the weighted complexity still weakly decreases. This yields a first blocker-sensitive theorem-state model: not all obligations are equally expensive, and reducing a hard blocker can matter disproportionately even before the local burden count is large.

6.5 Why This Matters Beyond Theorem Proving

The theorem case is not only a niche example. It is a clean case in which the broader phenomenon becomes visible. Research systems, scientific agents, and collective problem-solvers often behave similarly. They become stronger when they discover better objects to think *with*, not only more actions to try.

6.6 A Short `Nous` Reading

In the Nous setting, this means the main intelligence question is not merely whether the system produces more local proof attempts. It is whether it discovers:

- better local targets,
- better branch rotations,
- better blocker taxonomies,
- better reusable local gains,
- and better theorem-neighborhood structure.

Even in a theorem engine, raw throughput is only the visible surface. The deeper object is the system's ability to generate the representations that make proof search cheaper and more compositional.

---

7. Collective Intelligence As Externalized Cognitive Closure

7.1 Not More Workers, But Better Closure

The same logic clarifies collective intelligence. A collective does not become more intelligent merely because many local solvers exist side by side. It becomes more intelligent when representation, state, verification, and negative knowledge are externalized into a stable shared organization.

That organization can preserve and improve better ρ-choices over time:

- better state summaries,
- better decompositions,
- better blocker categories,
- better reusable intermediate objects,
- better transfer surfaces.

7.2 The Real Meaning Of Collective Gain

This is the deeper meaning of collective intelligence in the present framework. It is not raw multiplicity. It is the creation of an externalized cognitive closure that keeps lowering effective difficulty on future tasks rather than only generating more simultaneous guesses.

Cheap multi-agent systems therefore become smarter only when the organizational gain in state, modeling, verification, learning, and transfer exceeds the coordination cost. Otherwise the system becomes busier, not more intelligent.

---

8. Implications For Evaluation And Design

8.1 Evaluation

The framework implies that evaluation should move away from asking only:

- how high is the benchmark score,
- how many tasks were solved,

- how strong was the local worker.

Instead one should ask:

- did the difficulty frontier move,
- did the system lower effective difficulty through a better representation,
- did it discover reusable intermediate objects,
- did those gains persist into nearby tasks?

8.2 Design

The design implication is equally strong. If one wants a system to become more intelligent under budget, the goal should not be to search the same space harder by default. The goal should be to improve the system's ability to:

- discover better state variables,
- discover better decompositions,
- create better intermediate objects,
- preserve negative knowledge,
- and externalize those gains into a stable shared organization.

This is true for theorem engines, research systems, and collective AI designs more generally.

8.3 Why The Layered Machinery Still Matters

The earlier vector picture and emergence threshold have not been abandoned. They remain the supporting machinery of the paper. The five layers explain where systems fail. The emergence threshold explains when intelligence has appeared as an organized form. The new central claim says what many of those gains are *for*: they compress effective difficulty.

8.4 Research/Theorem Agent Design Principle

The theory yields a direct design principle for research and theorem agents:

> optimize first for representational leverage, not for raw proposal count.

In practical terms, the first question for a theorem or research agent should not be "what is the next token sequence to emit?" It should be:

- what representation makes the task easier,
- what intermediate object would collapse future work,
- what decomposition would reduce the dominant burden coordinate,
- what blocker taxonomy would prevent repeated false routes,
- and what reusable artifact should be preserved if a local gain is found.

This principle follows directly from Propositions 4-7. If intelligence is partly the organized reduction of effective difficulty, then agent design should prioritize the modules or loops that discover and preserve that reduction.

For theorem systems, the same principle can be written more sharply as a control objective:

> choose actions that maximize expected obligation-complexity reduction per unit cost

subject to truth preservation and verification constraints. This is the domain-specific reading of organized difficulty compression.

In a more proof-oriented implementation, this should be strengthened again:

> prefer intermediate objects whose value can be transported, at least approximately, from verified obligation simplification to active-burden relief.

That is the operational meaning of the formal interface in Section 6.4. The agent should not merely search for locally helpful artifacts. It should search for artifacts with certifiable downstream leverage.

8.5 A Minimal Evaluation Protocol For Theorem/Research Systems

The same framework yields a compact evaluation protocol for theorem and research agents. Under matched budget, evaluate whether the system improves the following quantities over a flat or representation-blind baseline:

1. **frontier lift**: does the system solve a harder task family under the same budget?
2. **representation lift**: does it find a formulation that lowers effective burden, measured through better route success or reduced obligation complexity?
3. **intermediate-object yield**: how often does it discover lemmas, invariants, bridge objects, or decomposition objects that survive verification?
4. **repeat-error suppression**: does negative knowledge reduce recurrence of failed routes?
5. **transfer reuse**: do earlier intermediate objects reduce cost on nearby tasks?
6. **obligation-compression rate**: how much verified remaining obligation complexity is removed per unit cost once a useful intermediate object is found?
7. **transport certification**: when an intermediate object helps, can the system explain or verify which active burden coordinate was relieved and by what monotone proxy?

This protocol is intentionally narrower than a universal intelligence benchmark. Its purpose is to test the paper's core thesis where it should be most visible: in theorem and research systems whose main advantage should come from structured difficulty compression rather than raw local throughput alone.

In a richer theorem-state implementation, these metrics should become blocker-sensitive: reducing one hard bridge or blocker obligation may matter more than shaving a small amount from many routine local rewrites.

8.6 A Formalization Ladder For Research/Theorem Systems

The paper now suggests a natural three-stage proof program.

1. **algebraic stage**: prove the order-theoretic core behind Propositions 4 and 6, plus the constructive witness behind Proposition 5;
2. **semantic theorem stage**: add the minimal theorem interface from Section 6.4 so that admissible intermediate objects transport obligation-complexity reduction into active-burden relief and then into frontier lift;
3. **empirical systems stage**: test whether real theorem or research agents actually satisfy useful proxies for the abstract burden and obligation quantities.

This ladder is important because it separates three questions that are often conflated:

- what is formally true inside the abstract model,
- what semantic assumptions are needed to connect the model to theorem work,
- and what measurements are needed to connect the theorem model to actual AI systems.

That separation should help the project grow in the right order. First prove the clean kernel. Then attach domain meaning. Then test the meaning against systems.

---

9. Conclusion

The most important correction proposed in this paper is simple:

> intelligence is not best understood as raw success inside a fixed problem space.

At system level, intelligence is better understood as:

- objective-relative,
- layered,
- emergent,
- and, centrally, **difficulty-compressing**.

Smarter systems often do not win by searching the same space harder. They win by finding a better space in which the task becomes easier for the same underlying budget. Better representations, better decompositions, better intermediate objects, and better shared cognitive structure reduce effective difficulty itself.

This is why theorem intelligence is such a revealing case. A theorem system becomes stronger not only when it proves more, but when it can discover the lemmas, invariants, and bridge objects that make proof obligations cheaper. The same pattern extends outward: collective intelligence is not merely many workers, but a stable externalized organization that preserves and improves the forms through which future work is done.

In that sense, intelligence is not only adaptive control. It is **organized difficulty compression**.