A Mathematical Theory of General Intelligence

State-Space Compression as a Variational Framework for Problem Solving

Daniel B. Austin, Ph.D.
Founder and CEO, Agent Logic, Inc.
doi:10.5281/zenodo.20738007

June 17, 2026

Abstract

We propose a mathematical framework that measures intelligence as the efficiency with which a system discovers and uses task-relevant state-space compressions under declared cost and capacity constraints. Starting from state-space compression (SSC), we define tasks, admissible representations, a resource-weighted loss functional, and Cognitive Compression Cost (CCC): the lowest achievable task cost under a declared modeling policy. An agent's task-local intelligence is its efficiency relative to this optimum, and general intelligence is the expectation of that score over a declared task distribution. We establish existence, rate-distortion, monotonicity, dominance, information-bottleneck, and no-free-lunch properties under explicit assumptions. The framework separates task difficulty from agent capability and makes resource assumptions visible, enabling conditional cross-substrate comparison when task, loss, representation, and resource policies are shared. It does not claim a theory of consciousness, personhood, or context-free intelligence ranking.

1 Introduction

How should we compare the intelligence of systems built from different substrates, architectures, and evolutionary histories? A honeybee navigating to a food source, a chess engine evaluating positions, a human scientist constructing a theory, and a hypothetical extraterrestrial intelligence solving problems we cannot yet imagine—each may be intelligent, but there is no consensus framework for comparing them under a shared account of task difficulty, resource cost, and substrate-neutral performance.

The difficulty is not merely that definitions of intelligence are vague. It is that most attempts to formalize intelligence begin from the wrong primitive. Benchmark performance conflates task difficulty with agent capability. Human-likeness tests are anthropocentric by construction. Reward-sum measures assume a specific agent-environment interaction model. Architecture-specific metrics cannot cross substrate boundaries.

A physicist confronting this problem would ask different questions: What is the state space? What is the observable? What does a good representation cost? How much of the microstate can be discarded while preserving what matters for prediction or action? These are the questions of state-space compression [23, 22].

This paper redirects SSC from scientific coarse-graining to intelligence measurement. The central claim is:

> Intelligence is the efficiency with which a system discovers and uses task-relevant state-space compressions under declared cost, measurement, and computational constraints.

The resulting framework is built from four ideas. First, a problem is specified by a task, observable, action surface, loss, horizon, and resource policy. Second, a representation is useful when it compresses the task state while preserving what is needed for prediction or action. Third, problem difficulty is modeled by the lowest achievable cost of such a representation inside a declared admissible class. Fourth, agent capability is modeled by comparing the cost an agent achieves against that task-relative optimum.

This framing is deliberately conditional. It does not claim an absolute measure independent of task, encoding, or cost model. Instead, it makes those choices explicit. Under shared assumptions, the framework enables comparison across substrates—not by asking whether two systems are built the same way, but by asking how efficiently each finds valid compressions of the same problem state.

The epistemological stance is Popperian. Popper argued that all life is problem solving and that knowledge grows through conjecture and refutation [15, 14]. In the present framework, an agent navigating representation space is doing exactly this: generating candidate compressions (conjectures), evaluating them against the task loss (refutations), and adopting lower-cost representations (knowledge growth). Intelligence, on this view, is the efficiency of this error-elimination process.

The paper proceeds as follows. Section 2 reviews SSC. Sections 3–4 define tasks, representations, and the variational objective. Section 5 defines Cognitive Compression Cost and proves existence, a rate-distortion lower bound, and structural results. Section 6 defines agent-relative intelligence and establishes axiomatic properties. Section 7 handles degenerate cases. Section 8 defines general intelligence over task distributions. Section 9 presents worked examples. Section 10 defines learning as a trajectory through representation space. Section 11 surveys related work, with detailed comparison to AIXI, Chollet's ARC framework, the information bottleneck, and minimum description length. Section 12 discusses limitations. Section 13 concludes.

2 Background: State-Space Compression

State-space compression, as developed by Wolpert, Libby, Grochow, and DeDeo [23], addresses the problem of choosing macrostates for complex systems. A fine-grained state $x_t \in X$ is mapped to a coarser state $y_t \in Y$ by a compression map $\pi : X \to Y$. The compressed state is useful when it can be measured, evolved, and used to predict an observable of interest at lower total cost than working directly with the fine-grained state.

The earlier SSC preprint [22] makes the purpose explicit: predict observables of a high-dimensional system accurately while minimizing computational cost. Compression is not merely shortening. A compression is valuable only relative to an observable, a loss, and a cost model. A good macrostate trades off measurement cost, computation cost, and prediction accuracy.

This paper adopts that structure but changes the application domain. Instead of asking how a scientist should coarse-grain a physical system, we ask how any problem-solving system should construct, refine, and use a representation of a task. This shift is what makes SSC a natural language for intelligence: it keeps the mathematics close to familiar questions about macrostates, observables, dynamics, and cost, while giving intelligence a definition that does not depend on human likeness or machine architecture.

The connection to related formalisms—rate-distortion theory [1, 3], the information bottleneck [21], computational mechanics [4, 18], and minimum description length [16, 5]—is discussed in Section 11.

3 Tasks and Representations

Definition 1 (Task)**.** *A task τ is a tuple*

$$\tau = (X, O, A, H, \ell, \kappa),$$

where X is a fine-grained state space, O is the observable or validation surface, A is an action or intervention surface, H is a time or interaction horizon, ℓ is a task loss, and κ is a resource policy.

A task might be predicting a physical observable, controlling a system, solving a mathematical problem, repairing a program, or navigating to a food source. What matters is not the domain but the presence of a declared observable, loss, and resource policy. The action surface A is included because intelligent systems must act, not merely predict; representations that are sufficient for forecasting may fail under intervention if they do not preserve causal structure [13].

Definition 2 (Representation)**.** *For a task τ, a representation r consists of a compression map $\pi_r : X \to Y_r$, an update or evolution rule over Y_r, and a prediction, control, or validation map from Y_r to the task observable or action surface.*

Informally: a representation is any structured reduction of state that preserves task-relevant distinctions while discarding what does not matter for the problem at hand.

Definition 3 (Admissible Representation Class)**.** *For a task τ, let $\mathcal{R}_\tau$ denote the set of representations permitted under the task's modeling policy. The admissible class may restrict measurement access, representation size, computational resources, permitted actions, memory, or search procedures.*

The admissible representation class is essential. Without it, claims about optimality are not meaningful. A representation that uses an oracle, unlimited computation, or hidden information should not be compared against an agent without access to those resources.

4 The Variational Objective

For a representation $r \in \mathcal{R}_\tau$, define a task-relative loss functional

$$\mathcal{L}(r \mid \tau) = E_\tau(r) + \lambda_c\, C_{\text{comp}}(r) + \lambda_m\, C_{\text{meas}}(r) + \lambda_r\, C_{\text{repr}}(r),$$

where $E_\tau(r)$ is residual task error (prediction error, control error, or validation failure), $C_{\text{comp}}(r)$ is computation cost, $C_{\text{meas}}(r)$ is measurement cost, and $C_{\text{repr}}(r)$ is representation maintenance cost. The nonnegative coefficients λ_c, λ_m, and λ_r express the declared resource policy.

Assumption 1 (Commensurability)**.** *The terms in $\mathcal{L}$ are normalized or otherwise made commensurable before the objective is used for comparison.*

Assumption 2 (Declared Resource Policy)**.** *The resource weights and admissible representation class are declared before evaluating any agent.*

The loss functional makes explicit what most evaluations leave implicit: the cost of measuring, representing, computing, and remaining wrong. The computation cost term is what separates this framework from pure information-theoretic measures: an encoding that is information-theoretically optimal may still be computationally prohibitive, and the framework captures this distinction.

A remark on units: the specific operationalization of each cost term (FLOPS, bits, wall-clock time, metabolic energy, number of steps) is a modeling choice within the resource policy κ. Different choices produce different CCC values, which is not a defect—it is the framework's insistence that cost assumptions be stated rather than hidden.

5 Cognitive Compression Cost

Definition 4 (Cognitive Compression Cost)**.** *The Cognitive Compression Cost of a task τ is*

$$\mathrm{CCC}(\tau) = \inf_{r \in \mathcal{R}_\tau} \mathcal{L}(r \mid \tau).$$

CCC is a task-relative notion of problem difficulty. It is defined only relative to the declared task encoding, observable, loss, resource policy, and admissible representation class. If $\mathcal{R}_\tau$ is broadened, CCC may decrease. If the resource policy penalizes measurement more heavily, CCC may increase. This conditionality is a feature: it forces intelligence claims to state the assumptions under which they are made.

Remark 1. *CCC separates two questions that are often conflated: how hard a task is under a declared policy, and how well a particular agent performs on that task.*

We now establish formal properties of CCC.

Theorem 1 (Existence of Optimal Compression)**.** *Let $\mathcal{R}_\tau$ be a compact set in a topology under which $\mathcal{L}(\cdot \mid \tau)$ is lower semi-continuous. Then the infimum defining $\mathrm{CCC}(\tau)$ is attained: there exists $r^* \in \mathcal{R}_\tau$ such that $\mathcal{L}(r^* \mid \tau) = \mathrm{CCC}(\tau)$.*

Proof. This is an application of the extreme value theorem for lower semi-continuous functions on compact sets. Since $\mathcal{L}(\cdot \mid \tau)$ is lower semi-continuous and $\mathcal{R}_\tau$ is compact, $\mathcal{L}$ attains its infimum on $\mathcal{R}_\tau$. □

Remark 2. *The compactness requirement should be read as a modeling condition, not as an automatic consequence of bounded resources. It is satisfied in finite admissible classes and in closed bounded parametric classes under an appropriate topology. When $\mathcal{R}_\tau$ is not compact (e.g., the class of all computable functions), the infimum may not be attained, and CCC should be interpreted as a greatest lower bound.*

Proposition 1 (Monotonicity Under Class Expansion)**.** *If $\mathcal{R} \subseteq \mathcal{R}'$ are two admissible representation classes for the same task τ, then*

$$\mathrm{CCC}(\tau \mid \mathcal{R}') \leq \mathrm{CCC}(\tau \mid \mathcal{R}).$$

Proof. The infimum over a larger set is at most the infimum over a subset. □

Remark 3. *This means that expanding the set of permitted representations can only reduce (or maintain) the difficulty of a task. Conversely, restricting the admissible class increases CCC, reflecting the intuition that tighter constraints make problems harder.*

Theorem 2 (Rate-Distortion Lower Bound on CCC)**.** *Consider a prediction task τ with task error $E_\tau(r) = \mathbb{E}[d(O, \hat{O}_r)]$ for a distortion measure d, where $O = f(X)$ is the observable. Suppose the admissible representation class $\mathcal{R}_\tau$ is constrained so that every $r \in \mathcal{R}_\tau$ satisfies $I(X; Y_r) \leq R$ for a rate bound $R > 0$, and that each representation induces an estimator $\hat{O}_r = \hat{O}_r(Y_r)$ of the observable. Let*

$$D(R) = \inf_{\substack{p(y|x): \\ I(X;Y) \leq R}} \mathbb{E}[d(O, \hat{O}(Y))]$$

be the distortion-rate function for observable O under the best possible stochastic map at rate R. Define the minimum operational costs $C^{\min}_{\mathrm{comp}} = \inf_{r \in \mathcal{R}_\tau} C_{\mathrm{comp}}(r)$ and $C^{\min}_{\mathrm{meas}} = \inf_{r \in \mathcal{R}_\tau} C_{\mathrm{meas}}(r)$, and assume $C_{\mathrm{repr}}(r) \geq 0$ for all admissible r. Then

$$\mathrm{CCC}(\tau) \;\geq\; D(R) + \lambda_c \, C^{\min}_{\mathrm{comp}} + \lambda_m \, C^{\min}_{\mathrm{meas}}.$$

Proof. For any $r \in \mathcal{R}_\tau$, the representation's induced channel $X \mapsto Y_r$ and estimator $\hat{O}_r(Y_r)$ form a feasible candidate in the distortion-rate problem at rate at most R. Therefore $E_\tau(r) \geq D(R)$ by definition of the distortion-rate function. Therefore

$$\begin{aligned}\mathcal{L}(r \mid \tau) &= E_\tau(r) + \lambda_c\, C_{\text{comp}}(r) + \lambda_m\, C_{\text{meas}}(r) + \lambda_r\, C_{\text{repr}}(r) \\ &\geq D(R) + \lambda_c\, C_{\text{comp}}(r) + \lambda_m\, C_{\text{meas}}(r) \\ &\geq D(R) + \lambda_c\, C_{\text{comp}}^{\min} + \lambda_m\, C_{\text{meas}}^{\min}.\end{aligned}$$

Since this holds for every $r \in \mathcal{R}_\tau$, it holds for the infimum. □

Corollary 1 (Gaussian Prediction Bound)**.** *If O is a scalar Gaussian observable with variance σ^2, d is squared error, and the rate constraint is R bits, then*

$$\text{CCC}(\tau) \;\geq\; \sigma^2\, 2^{-2R} + \lambda_c\, C_{\text{comp}}^{\min} + \lambda_m\, C_{\text{meas}}^{\min}.$$

Proof. For a Gaussian source under squared-error distortion, the distortion-rate function is $D(R) = \sigma^2\, 2^{-2R}$ [3]. Substitution into Theorem 2 gives the result. □

Remark 4. *Theorem 2 establishes that information-theoretic limits on compression quality translate directly into lower bounds on CCC. The bound decomposes into an irreducible distortion floor set by rate-distortion theory and an irreducible operational floor set by the cheapest available measurement and computation. Under the stated task, rate constraint, admissible class, and cost model, no agent can beat this bound regardless of architecture or substrate. The Gaussian corollary shows that for well-studied source distributions, the bound produces concrete numerical values, making CCC amenable to quantitative analysis rather than purely qualitative comparison.*

6 Agent-Relative Intelligence

Let $\mathcal{L}_\tau^*(a)$ denote the best task loss achieved by agent a on task τ under the same evaluation policy used to define CCC.

Definition 5 (Task-Local Intelligence Score)**.** *For $\text{CCC}(\tau) > 0$ and $\mathcal{L}_\tau^*(a) > 0$, define*

$$I(a, \tau) = \frac{\text{CCC}(\tau)}{\mathcal{L}_\tau^*(a)}.$$

Proposition 2 (Bounded Ratio)**.** *If $\mathcal{L}_\tau^*(a) \geq \text{CCC}(\tau) > 0$, then $0 < I(a, \tau) \leq 1$.*

Proof. By assumption, $\mathcal{L}_\tau^*(a) \geq \text{CCC}(\tau) > 0$. Dividing gives $I(a, \tau) \leq 1$; positivity of both terms gives $I(a, \tau) > 0$. □

Remark 5. *If an agent obtains $I(a, \tau) > 1$, the result is diagnostic: the admissible class was too narrow, the CCC estimate was incorrect, or the agent was evaluated under a different resource policy.*

7 Conventions for Degenerate Cases

A task τ with $\mathrm{CCC}(\tau) = 0$ is *null* and excluded from ratio scoring. A task is *infeasible* for $\mathcal{R}_\tau$ when no admissible representation meets the minimum validation condition; infeasible tasks are reported rather than scored. A *failed* agent run (one that does not produce a valid trace) is reported as failure, not as low intelligence. For stochastic agents, the score is reported as a distribution or declared statistic (e.g., expected value or quantile):

$$\bar{I}(a, \tau) = \mathbb{E}_\omega \left[\frac{\mathrm{CCC}(\tau)}{\mathcal{L}^*_\tau(a, \omega)} \right],$$

when the expectation exists and failures are handled by an explicit censoring policy.

8 General Intelligence Over a Task Distribution

Task-local intelligence is not general intelligence. To define a broader measure, let $\mathcal{D}$ be a distribution over tasks together with a declared scoring policy for null, infeasible, failed, and stochastic runs. Let $\mathcal{S}_{a,\mathcal{D}}$ denote the tasks in the support of $\mathcal{D}$ for which the policy yields a positive, finite task-local score for agent a. By default, null and infeasible tasks are excluded from the scored distribution and reported separately; failed runs are not converted into low intelligence unless the policy explicitly assigns a finite penalty.

Definition 6 (General Intelligence)**.** *When $I(a, \tau)$ is defined and integrable on the scored task set, define*

$$G(a; \mathcal{D}) = \mathbb{E}_{\tau \sim \mathcal{D} | \tau \in \mathcal{S}_{a,\mathcal{D}}}[I(a, \tau)].$$

This makes the benchmark distribution explicit. Two agents may be ranked differently under different task distributions. That is not a defect; it is a feature. General intelligence claims should state the family of problems over which they are made, the scoring policy used for edge cases, and the mass of tasks excluded or censored by that policy.

Definition 7 (Comparative General Intelligence)**.** *Agent a is more generally intelligent than agent b over $\mathcal{D}$ when $G(a; \mathcal{D}) > G(b; \mathcal{D})$ under a common scored task set and the same declared scoring policy.*

This is the comparison primitive. It is not a claim about moral status, consciousness, or social worth. It is a claim about which system solves the declared family of problems at lower task cost. If two agents induce different scored task sets because of null, infeasible, failed, or censored runs, the comparison is undefined until the evaluation policy supplies a common scored set or an explicit finite penalty/censoring rule.

Proposition 3 (Dominance Ordering)**.** *Assume agents a and b are evaluated under the same task distribution and scoring policy, with a common scored task set of positive measure. If agent a achieves $\mathcal{L}^*_\tau(a) \leq \mathcal{L}^*_\tau(b)$ for every scored task τ in the support of $\mathcal{D}$, then $G(a; \mathcal{D}) \geq G(b; \mathcal{D})$. If the inequality is strict on a set of positive measure under $\mathcal{D}$, then $G(a; \mathcal{D}) > G(b; \mathcal{D})$.*

Proof. For each scored τ with $\mathrm{CCC}(\tau) > 0$, the condition $\mathcal{L}^*_\tau(a) \leq \mathcal{L}^*_\tau(b)$ implies $I(a, \tau) \geq I(b, \tau)$. Taking expectations over the common scored distribution preserves the inequality. Strict inequality on a positive-measure set produces strict inequality in the expectation. □

Proposition 4 (No Universal Fixed Search Strategy)**.** *Let $\mathcal{Q}$ contain all fixed-order search procedures over a shared set of $n \geq 2$ candidate terminal representations. Each procedure incurs a per-evaluation cost $c > 0$ before using the first optimal representation it finds. For this proposition, $\mathcal{L}$ includes both the terminal representation cost and the search cost, so $\mathrm{CCC}(\tau)$ is the infimum over admissible procedures in $\mathcal{Q}$. Let $\mathcal{T}$ contain, for each terminal representation, at least one task whose unique optimal terminal representation is that representation. Then no fixed search procedure achieves $I(q, \tau) = 1$ for all $\tau \in \mathcal{T}$.*

Proof. An agent achieves $I(a, \tau) = 1$ only when its total cost equals $\mathrm{CCC}(\tau)$ under the admissible procedure class; that is, when its search order finds the optimal terminal representation with minimum total search and terminal-representation cost.

Fix any procedure q with evaluation order $r_{q(1)}, r_{q(2)}, \ldots, r_{q(n)}$. Because $n \geq 2$ and $\mathcal{T}$ contains at least one task for each representation, there exists a task τ whose unique optimal representation is not $r_{q(1)}$. For that task, procedure q must evaluate at least two representations before finding the optimum, incurring search cost $\geq 2c$.

The minimum possible search cost for that same task is c, achieved by a procedure that evaluates $r^*(\tau)$ first. Since search cost is included in $\mathcal{L}$ and $\mathrm{CCC}(\tau)$ is the optimum over admissible procedures, procedure q's achieved cost is strictly larger than the task optimum: $\mathcal{L}^*_\tau(q) > \mathrm{CCC}(\tau)$. Hence $I(q, \tau) < 1$ for that task. Because the choice of q was arbitrary, no fixed search procedure achieves $I(q, \tau) = 1$ for every $\tau \in \mathcal{T}$.

This is the no-free-lunch content of the result [24]: without additional structure that privileges some optima over others, no fixed search order is universally best. The result does not say that search procedures cannot be useful. It says that universal optimality is unavailable without task structure. □

Remark 6. *This result formalizes why general intelligence must be defined relative to a task distribution $\mathcal{D}$ rather than over all possible tasks. A search strategy can be well-adapted to a particular distribution—placing likely optima early in its evaluation order—but no strategy dominates across all distributions simultaneously. The choice of $\mathcal{D}$ is therefore an inescapable part of any general intelligence claim. This is the CCC analogue of Wolpert and Macready's central message: performance advantages on one class of problems are always paid for by disadvantages on another.*

9 Worked Examples

9.1 Parity Extraction

Consider a task where the state is a binary vector $x \in \{0,1\}^{1000}$ and the observable is the parity of the first ten bits: $O(x) = \left(\sum_{i=1}^{10} x_i\right) \bmod 2$. The resource policy assigns one unit of cost per measured bit, one unit per stored bit, one unit per binary operation, and a penalty of 100 for an incorrect answer:

$$\mathcal{L}(r \mid \tau) = 100\, E_\tau(r) + C_{\mathrm{meas}}(r) + C_{\mathrm{repr}}(r) + C_{\mathrm{comp}}(r).$$

A microstate representation reads all 1000 bits, stores them, and computes the parity: cost $0 + 1000 + 1000 + 9 = 2009$. A task-relevant compression reads only the first ten bits, stores one parity bit, and performs nine XOR operations: cost $0 + 10 + 1 + 9 = 20$. Under a class admitting both, $\mathrm{CCC}(\tau) \leq 20$.

If the admissible class requires exact parity computation, charges only the declared measurement, storage, and binary-operation costs, and contains no cheaper circuit or oracle-like shortcut, this establishes $\mathrm{CCC}(\tau) = 20$ within the toy policy.

Compare three agents. Agent a_1 discovers the parity representation: $\mathcal{L}^*_\tau(a_1) = 20$. Agent a_2 reads the full microstate: $\mathcal{L}^*_\tau(a_2) = 2009$. Agent a_3 reads five bits and guesses: $\mathcal{L}^*_\tau(a_3) = 50+5+1+4 = 60$.

$$I(a_1, \tau) = 1, \quad I(a_2, \tau) \approx 0.010, \quad I(a_3, \tau) \approx 0.333.$$

Agent a_2 has all the information and reaches the right answer, yet it is far less intelligent on this task because it fails to find the compression. Agent a_3 uses less measurement but pays for residual error. The comparison is about compression efficiency, not substrate or architecture.

9.2 Thermostat Control

Consider a room with microstate x encoding molecular positions, wall insulation properties, heating element state, and furniture thermal mass. The task observable is $O = |T_{\text{room}} - T_{\text{set}}|$ (temperature deviation from setpoint), the action surface is $A = \{\text{heater on, heater off}\}$, the horizon is one hour, and the loss penalizes deviation, energy use, and switching cost.

A minimal representation r_1 tracks only current room temperature and heating rate—two real numbers. Its compression map discards all microstate detail irrelevant to the control task. Cost: low measurement (two sensors), low representation (two floats), low computation (threshold comparison).

A fluid-dynamics representation r_2 models convection, conduction, and radiation from the full microstate. It achieves marginally lower control error but at vastly higher measurement, representation, and computation cost.

Under a policy that heavily weights measurement and computation while assigning only modest value to the marginal control-error reduction, r_1 dominates r_2 for this task: it achieves nearly the same E_τ at a fraction of the other costs. An agent that discovers r_1 scores near $I = 1$. An agent that insists on r_2 may be a better physicist but is a less intelligent thermostat controller.

This example illustrates the action surface. The representation must support not just prediction of temperature but selection of the correct heating action. A representation that predicts temperature perfectly but cannot determine when to switch the heater is useless for this task, even if it achieves zero prediction error.

9.3 Cross-Substrate Comparison: Foraging

Consider a foraging task on a 100×100 grid with obstacles and a food source. The observable is food acquisition within a step budget. The action surface is {north, south, east, west, stay}. The resource policy charges one unit per step and one unit per unit of internal computation.

Agent a_{bio} is a biological organism using spatial memory: it forms a landmark-based representation of the grid, remembers approximate food locations, and navigates by dead reckoning. Agent a_{sw} is a software agent using A^* search on a full grid map.

Both solve the task. Agent a_{bio} uses a compressed representation (landmarks, approximate distances) that requires little memory but incurs some navigation error. Agent a_{sw} uses a complete grid map with optimal pathfinding but pays high representation and computation costs.

Under a resource policy that charges equally for steps and computation, a_{bio} may achieve a lower total $\mathcal{L}$ on most grid configurations despite being suboptimal in path length. The framework compares these agents without requiring substrate equivalence: both are evaluated against the same task tuple, loss functional, and resource policy.

The comparison is conditional—change the resource policy to penalize steps more than computation, and a_{sw}'s optimal paths may win. This conditionality is the framework's strength: it prevents the question "which is smarter?" from receiving a context-free answer.

9.4 Random vs. Structured Prediction

Let $X = \{0,1\}^{100}$ with uniform i.i.d. dynamics. Consider two tasks on the same state space.

Task A: predict the exact next state. No compression can reduce residual error below $1 - 2^{-100}$; because the next state is independent of the current state, storing the current microstate does not help. CCC is dominated by the unavoidable error penalty and is high under exact-state loss.

Task B: predict the empirical frequency of 1s over the next 1000 steps. The sufficient statistic is a running count. A one-number representation achieves low prediction error at minimal cost. CCC is low.

This demonstrates that CCC is a property of the task, not the system. The same state-space dynamics yield radically different difficulty depending on the observable and loss. Claims that a system is "simple" or "random" are incomplete without specifying what one is trying to predict or control.

9.5 Learning Trajectory: Compression Improvement Over Episodes

This example illustrates the learning framework of Section 10 with a concrete agent improving its representation over successive episodes.

Consider a repeated prediction task. At each episode $t = 1, 2, \ldots, T$, the agent observes a sequence of 100 symbols drawn from $\{A, B, C, D\}$ and must predict the next symbol. The generating process has hidden structure: the sequence is generated, for purposes of this toy policy, by a first-order Markov chain with a 4×4 transition matrix M that is sparse (most entries are zero) and stationary across episodes. The resource policy charges 1 unit per symbol stored in memory, 1 unit per parameter maintained, and 10 units per incorrect prediction.

Under an admissible class restricted to Markov models over the four symbols, the optimal representation is the transition matrix M itself: 16 parameters, of which only (say) 6 are nonzero. A representation storing only the 6 nonzero entries achieves near-perfect prediction. Suppose the declared finite-sample estimation model assigns this sparse representation a residual error rate of approximately 5% with 100 symbols per episode. Under those toy assumptions, the optimal cost is $\mathrm{CCC}(\tau) = 6 + 10 \cdot 0.05 = 6.5$.

Now consider an agent that learns. At episode 1, the agent uses a naive representation r_1: it memorizes raw symbol frequencies (a 4-element histogram). This captures marginal statistics but misses transition structure. Achieved cost: $\mathcal{L}(r_1 \mid \tau) = 4 + 10 \cdot 0.6 = 10$ (four parameters, 60% prediction error).

At episode 3, after observing transition patterns, the agent adopts r_2: the full 16-entry transition matrix estimated from data. Achieved cost: $\mathcal{L}(r_2 \mid \tau) = 16 + 10 \cdot 0.05 = 16.5$ (better prediction but high representation cost from maintaining zero entries).

At episode 7, the agent prunes zero entries and adopts r_3: the sparse 6-parameter transition model. Achieved cost: $\mathcal{L}(r_3 \mid \tau) = 6 + 10 \cdot 0.05 = 6.5 = \mathrm{CCC}(\tau)$.

The trajectory in representation space is:

$$\mathcal{L}(r_1 \mid \tau) = 10 \xrightarrow{C_{\text{adapt}}} \mathcal{L}(r_2 \mid \tau) = 16.5 \xrightarrow{C_{\text{adapt}}} \mathcal{L}(r_3 \mid \tau) = 6.5.$$

The intelligence scores evolve non-monotonically: $I_1 \approx 0.65$, then $I_2 \approx 0.39$ (a temporary regression as the agent overfits its representation), then $I_3 = 1$. The cumulative learning cost includes the adaptation penalties $C_{\text{adapt}}(r_1, r_2)$ and $C_{\text{adapt}}(r_2, r_3)$ for restructuring the internal model. In the notation introduced below, $\mathcal{J}(a)$ is the total cost of this trajectory: the achieved costs along the path plus the adaptation penalties paid to move between representations.

Several features of this trajectory are worth noting. First, the path is non-monotonic: r_2 is temporarily worse than r_1 in total cost despite being more accurate, because it over-represents the state. This is a common pattern in learning systems—acquiring more detailed models before discovering which details can be discarded. Second, the final compression r_3 is not merely a reduction of r_2; it reflects a qualitative insight (sparsity) that r_2 does not contain. Third, a more intelligent learner might skip r_2 entirely, moving directly from frequency counts to sparse transition models—a faster trajectory through representation space with lower cumulative learning cost.

In Popperian terms, r_1 is a weak conjecture (marginal frequencies), r_2 is a richer but wasteful conjecture (full transition matrix), and r_3 is the refined conjecture that survives error elimination (sparse sufficient statistics). The agent's learning intelligence is measured not just by reaching r_3, but by how efficiently it navigates the path.

10 Learning as Trajectory in Representation Space

The static framework compares agents at a single point in time. Learning extends this to trajectories.

Definition 8 (Representation Trajectory)**.** *A representation trajectory for agent a over a task sequence $\tau_1, \tau_2, \ldots$ is a sequence of representations $r_1, r_2, \ldots$ adopted by the agent, where $r_t \in \mathcal{R}_{\tau_t}$.*

Definition 9 (Cumulative Learning Cost)**.**

$$\mathcal{J}(a) = \sum_{t=1}^{T} \mathcal{L}(r_t \mid \tau_t) + \lambda_a \sum_{t=2}^{T} C_{\text{adapt}}(r_{t-1}, r_t),$$

where $C_{\text{adapt}}(r_{t-1}, r_t)$ is the cost of transitioning from representation r_{t-1} to r_t (retraining, restructuring, forgetting).

Learning is thus modeled as movement through representation space toward lower-cost compressions, with an explicit penalty for the transition itself. An intelligent learner does not merely converge to good representations; it converges efficiently, without excessive adaptation cost.

This connects the framework to the literature on bounded rationality [19] and resource-rational cognition [12]: learning is the process by which an agent improves its compression strategy under resource constraints. It also connects to reinforcement learning [20], where the agent's policy can be viewed as a compressed representation of the value function.

In Popperian terms, the learning trajectory is a sequence of conjectures and refutations in representation space. Each new representation is a conjecture about what compression is useful; the task loss is the refutation signal; and the adoption of a lower-cost representation is knowledge growth [14].

11 Related Work

11.1 State-Space Compression and Information Theory

The direct parent of this paper is the SSC framework of Wolpert, Libby, Grochow, and DeDeo [23, 22]. We adopt their mathematical posture—compression maps, macrostate dynamics, observables, cost tradeoffs—and redirect it from scientific modeling to intelligence measurement. Kolchinsky and Wolpert's work on semantic information [9] is adjacent: it formalizes information that matters for an agent's viability, connecting to our notion of task-relevant compression.

Rate-distortion theory [1, 3] provides the foundational mathematics for lossy compression under distortion constraints. The CCC framework can be seen as an extension of rate-distortion to include computation cost, measurement cost, and task-specific validation—dimensions absent from the classical formulation.

Computational mechanics [4, 18] identifies causal states as minimal sufficient statistics for prediction. The ϵ-machine is, in CCC terms, an optimal representation for next-symbol prediction under a specific loss and representation class.

11.2 The Information Bottleneck

The information bottleneck (IB) method [21] finds a compressed representation T of input X that preserves information about target Y by minimizing $I(X;T) - \beta\, I(T;Y)$. The structural similarity to CCC is clear: both seek compressed representations that trade off compression against task relevance.

Proposition 5 (IB as Special Case)**.** *Consider a prediction task where the loss is mutual-information loss, $E_\tau(r) = H(O \mid Y_r)$, and where measurement cost and computation cost are zero ($\lambda_m = \lambda_c = 0$) and representation cost is measured by $I(X;Y_r)$. Then*

$$\mathrm{CCC}(\tau) = \inf_{Y_r} \left[H(O \mid Y_r) + \lambda_r\, I(X;Y_r) \right],$$

which is the IB optimum for trade-off parameter $\beta = 1/\lambda_r$.

Proof. Direct substitution into $\mathcal{L}$ under the stated cost assignments. □

CCC differs from IB by including measurement cost (not all variables are freely observable), computation cost (not all representations are equally cheap to update), and by admitting non-information-theoretic loss functions (control error, validation failure, causal fidelity). These additional cost dimensions are precisely what is needed for cross-substrate intelligence comparison, where the physical cost of acquiring and processing information varies between substrates.

11.3 Minimum Description Length

MDL [16, 5] evaluates hypotheses by the total description length of model plus data given model. The MDL principle and CCC share the intuition that compression quality indicates model quality. However, the two measures answer different questions. MDL asks: *what is the shortest description of this data?* CCC asks: *what is the cheapest useful representation for this task?* The word "useful" carries the distinction. A shortest description may be computationally intractable to decode, interpret, or act upon; CCC penalizes this through C_{comp}. A shortest description may require observing the entire state; CCC penalizes this through C_{meas}. MDL measures static encoding parsimony; CCC measures operational problem-solving cost.

Kolmogorov complexity [11] provides the theoretical lower bound on description length: the length of the shortest program that produces a given string. It is uncomputable but foundational. The distinction from CCC is again one of purpose. Kolmogorov complexity is substrate-independent but task-independent—it measures the intrinsic information content of an object regardless of what anyone wants to do with it. CCC is both substrate-independent and task-dependent: it measures the cost of a representation that is adequate for a declared purpose under declared constraints. A random bit string has high Kolmogorov complexity but may have low CCC if the task only requires predicting its frequency statistics.

11.4 AIXI and Universal Intelligence

Hutter's AIXI [8] defines an optimal agent that maximizes expected reward over all computable environments, weighted by the Solomonoff prior $2^{-K(\mu)}$ where K is Kolmogorov complexity. Legg and Hutter [10] use this to define a universal intelligence measure: $\Upsilon(\pi) = \sum_\mu 2^{-K(\mu)} V_\mu^\pi$, where V_μ^π is the expected reward of policy π in environment μ.

The CCC framework differs from AIXI in three respects.

First, **the primitive differs**. AIXI begins with reward maximization over environments; CCC begins with compression cost over representations. Intelligence in AIXI is total reward; intelligence in CCC is compression efficiency.

Second, **the environment weighting differs**. AIXI's Solomonoff prior is fixed and uncomputable; CCC's task distribution $\mathcal{D}$ is declared and operational. This makes CCC less universal but more usable: the comparison is valid under stated assumptions rather than under an implicit universal prior.

Third, **resource costs are explicit**. AIXI assumes unbounded computation; CCC charges for it. This matters for cross-substrate comparison: a biological system with low energy budget and a digital system with high computation budget face different cost landscapes, and CCC captures this while AIXI does not.

Under specific conditions (task = environment prediction, loss = negative reward, representation class = computable functions, zero resource costs), a CCC-optimal agent would agree with an AIXI-optimal agent. But the CCC framework exposes the resource costs that AIXI's unbounded computation hides, making it more appropriate for comparing physically realized systems.

Hernández-Orallo and collaborators [7, 6] develop formal intelligence measures that are computable and applicable to diverse agent types. Their work shares our goal of substrate-neutral comparison and provides a complementary approach grounded in algorithmic information theory.

Schmidhuber's compression-based intelligence measures [17] identify intelligence with compression progress—the discovery of novel regularities. This is compatible with the CCC framework: an agent that discovers a new compression for a task is making compression progress, and the resulting drop in $\mathcal{L}$ corresponds to intelligence gain. The difference is that CCC grounds the comparison in a declared task and resource policy rather than in a subjective notion of interestingness.

11.5 Chollet's ARC Framework

Chollet [2] argues that intelligence should be measured as skill-acquisition efficiency: how quickly a system generalizes to novel tasks given limited experience, with explicit priors and curriculum. The ARC benchmark operationalizes this through abstract visual reasoning tasks designed to be hard for pattern-matching and easy for systems with genuine generalization.

CCC and Chollet's framework share two key commitments: that intelligence claims must state the task distribution over which they are made, and that single benchmarks are insufficient. The frameworks are complementary rather than competing. Chollet measures how fast an agent *learns* to compress (skill acquisition efficiency); CCC measures how well an agent *compresses* (representation cost under declared policy). Chollet's framework emphasizes the transition from experience to capability; CCC emphasizes the quality of the resulting compression. A complete account of intelligence likely needs both: the efficiency of learning and the efficiency of the representations learned.

11.6 Bounded Rationality and Resource-Rational Cognition

Simon [19] argued that rational agents are bounded by cognitive resources and must satisfice rather than optimize. Lieder and Griffiths [12] formalize this as resource-rational analysis: cognition is the optimal use of limited computational resources.

CCC formalizes the same intuition within a compression framework. The resource policy κ in the task definition is the CCC analogue of Simon's bounds and Lieder–Griffiths's computational budgets. The key addition is that CCC provides a notion of task-relative optimality (the infimum of the loss functional) against which bounded agents can be measured.

11.7 Popper: Intelligence as Efficient Error Elimination

Popper's epistemology [14, 15] holds that knowledge grows through conjecture and refutation: we propose theories, test them, and retain those that survive criticism. His claim that "all life is problem solving" [15] motivated the present framework.

The connection is more than metaphorical. In CCC terms, a representation is a conjecture about what compression of the world is useful for a given task. The task loss is the refutation signal: it measures how badly the conjecture fails. Replacing a high-cost representation with a lower-cost one is knowledge growth. Intelligence, then, is the efficiency with which a system carries out this Popperian error-elimination process over representations.

This gives the framework an epistemological grounding distinct from purely information-theoretic measures. Rate-distortion theory and IB measure the cost of encoding; CCC measures something closer to the *cost of knowledge*—the total price of discovering, maintaining, and using a valid compression of a problem.

12 Limitations

The framework has important limitations.

First, CCC is not encoding-free. It depends on the declared task, observable, loss, and representation class. Different encodings of the same problem may yield different CCC values.

Second, the measure is not assumption-free. Its claim to generality comes from substrate neutrality, not from escaping task choice. It can compare systems when they are evaluated against the same task family and resource policy. It cannot provide meaningful comparison when those policies are hidden or inconsistent.

Third, cross-substrate comparison is the framework's goal, but it is a conditional goal. The framework makes such comparison *possible* by translating different systems into the same task-cost language. It does not produce a universal ranking independent of context.

Fourth, the computation cost term C_{comp} requires operationalization that may differ between substrates. Comparing FLOPS to metabolic energy to enzymatic reaction rates requires commensurability assumptions that the framework exposes but does not resolve.

Fifth, this paper proposes a framework. It does not provide empirical validation across benchmark suites, biological systems, or deployed agents. Estimation of CCC for realistic tasks and comparison of real agents remain essential future work.

Empirically, CCC can be estimated in bounded settings without claiming exact attainment. For finite discrete representation classes, one can evaluate all admissible representations directly. For parametric classes, optimization of $\mathcal{L}(r \mid \tau)$ supplies upper bounds on CCC, while rate-distortion or relaxation arguments can provide lower bounds, making the remaining uncertainty explicit.

Finally, the framework does not address consciousness, qualia, subjective experience, or moral status. A system may be highly intelligent under CCC without being conscious, and the framework is silent on whether intelligence implies or requires subjective experience.

13 Conclusion

This paper has proposed a mathematical framework for intelligence as compression-efficient problem solving. Starting from state-space compression, we defined tasks, admissible representations, a cost-weighted loss functional, and Cognitive Compression Cost as a task-relative measure of problem difficulty. Agent intelligence is the ratio of CCC to achieved cost. General intelligence is the expectation of this ratio over a declared task distribution.

We proved that CCC is attained under natural compactness conditions (Theorem 1), that it is bounded below by the distortion-rate function plus irreducible operational costs (Theorem 2), that it decreases monotonically under class expansion (Proposition 1), that the intelligence ratio respects dominance ordering (Proposition 3), that it reduces to the information bottleneck in a natural special case (Proposition 5), and that a no-free-lunch-style argument rules out one fixed search order being optimal across the constructed task family (Proposition 4). The rate-distortion bound, in particular, connects CCC to the quantitative machinery of information theory: for well-studied source distributions, it yields concrete numerical lower bounds on task difficulty (Corollary 1).

The framework's central benefit is separation. Problem difficulty lives in CCC; agent capability lives in the achieved cost. Resource assumptions live in the loss functional; comparison validity lives in the shared task distribution. This separation makes it possible, in principle, to compare problem-solving systems across substrates—not by asking whether they resemble each other, but by asking how efficiently each compresses the same problem under the same constraints.

The framework's central limitation is conditionality. Every comparison requires declared tasks, declared costs, and declared representation classes. This is not a defect: it is the framework's insistence that intelligence claims be honest about their assumptions. A comparison that hides its assumptions is not more general; it is less trustworthy.

Future work includes empirical estimation of CCC for concrete task families, design of task distributions that serve as meaningful intelligence benchmarks, formal analysis of CCC under encoding transformations, connections to thermodynamic limits on computation, and application to the evaluation and comparison of AI systems.

References

[1] Toby Berger. *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[2] François Chollet. On the measure of intelligence, 2019.

[3] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, Hoboken, NJ, 2 edition, 2006.

[4] James P. Crutchfield. The calculi of emergence: Computation, dynamics and induction. *Physica D*, 75(1–3):11–54, 1994.

[5] Peter D. Grünwald. *The Minimum Description Length Principle*. MIT Press, Cambridge, MA, 2007.

[6] José Hernández-Orallo. *The Measure of All Minds: Evaluating Natural and Artificial Intelligence*. Cambridge University Press, Cambridge, 2017.

[7] José Hernández-Orallo and David L. Dowe. Measuring universal intelligence: Towards an anytime intelligence test. *Artificial Intelligence*, 174(18):1508–1539, 2010.

[8] Marcus Hutter. *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability*. Texts in Theoretical Computer Science. An EATCS Series. Springer, Berlin, Heidelberg, 2005.

[9] Artemy Kolchinsky and David H. Wolpert. Semantic information, autonomous agency and non-equilibrium statistical physics. *Interface Focus*, 8(6):20180041, 2018.

[10] Shane Legg and Marcus Hutter. Universal intelligence: A definition of machine intelligence. *Minds and Machines*, 17(4):391–444, 2007.

[11] Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, New York, 3 edition, 2008.

[12] Falk Lieder and Thomas L. Griffiths. Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences*, 43:e1, 2020.

[13] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2 edition, 2009.

[14] Karl Popper. *Conjectures and Refutations: The Growth of Scientific Knowledge*. Routledge, London, 1963.

[15] Karl Popper. *All Life is Problem Solving*. Routledge, London, 1999.

[16] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[17] Jürgen Schmidhuber. Ultimate cognition à la Gödel. *Cognitive Computation*, 1(2):177–193, 2009.

[18] Cosma Rohilla Shalizi and Cristopher Moore. What is a macrostate? subjective observations and objective dynamics, 2003.

[19] Herbert A. Simon. A behavioral model of rational choice. *The Quarterly Journal of Economics*, 69(1):99–118, 1955.

[20] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2 edition, 2018.

[21] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method, 2000.

[22] David H. Wolpert, Joshua A. Grochow, Eric Libby, and Simon DeDeo. Optimal high-level descriptions of dynamical systems, 2014.

[23] David H. Wolpert, Eric Libby, Joshua A. Grochow, and Simon DeDeo. The many faces of state space compression. In Sara Imari Walker, Paul C. W. Davies, and George F. R. Ellis, editors, *From Matter to Life: Information and Causality*, pages 199–243. Cambridge University Press, Cambridge, 2017.

[24] David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.