A Theory of Intelligence as Energy-Efficient Cost Reduction: Competence, Learning, and Amortised Deliberation

April 29, 2026

Abstract

We define intelligence as the rate at which a system reduces the expected cost of performing tasks per unit of energy consumed. The choice of energy as denominator is the central structural commitment of the framework: unlike experience, data exposure, samples, or curriculum steps, energy is identically defined across biological, mechanical, and computational substrates and requires no substrate-specific indexing convention to be measured. Existing efficiency-based accounts of intelligence — notably Chollet's skill-acquisition efficiency and the Legg–Hutter universal-intelligence framework — index efficiency by experience or environment-weighted expectation rather than by a substrate-neutral physical resource, and offer no clean treatment of inference-time computation in modern AI systems. We propose energy as a substrate-neutral denominator and amortised deliberation as the corresponding treatment of inference-time computation. The energy choice is structurally significant: energy is exogenous to the learning process, whereas priors and experience are themselves outputs of prior learning, so experience-indexed frameworks measure efficiency relative to a reference state that is itself an output of the process being measured. The framework formally separates competence (a level — current task cost), learning (a process — cost reduction over an energy interval), and intelligence (a rate — cost reduction per unit of energy consumed across tasks). Intelligence values in this framework are well-defined relative to a specification of system boundary, task distribution, and cost functional; within a specification, cross-system comparison is substrate-neutral. Two consequences follow. First, a deployed frozen model has zero intelligence in this sense regardless of its capability, while the training system that produced it has positive intelligence; the artifact and the process that generates it are different objects measured by different quantities. Second, inference-time reasoning along a fixed-state performance frontier does not constitute intelligence, but distillation of inference products into the behavioural state through the update operator does — a composite cycle we call amortised deliberation. We prove structural properties of the definitions, discuss pathologies of narrow task distributions and threshold-based forms, and recommend refined empirical variants including transfer intelligence and proportional learning efficiency.

Keywords

Intelligence; Intelligence definitions; Energy-efficient learning; Substrate-neutral comparison; Artificial general intelligence; Thermodynamics of computation; Inference-time reasoning; Skill-acquisition efficiency

Contents

1 The Usual Denominator Problem

Intelligence has often been characterised as efficiency: a system is intelligent insofar as it reduces task cost per unit of some resource. The substantive question is which resource. Experience, data exposure, sample count, training steps, or curriculum interactions are each defensible candidates, and each faces the same structural problem: there is no substrate-neutral way to count them. What constitutes "one experience" for a continuous-time biological organism, a discrete-step reinforcement learner, a transformer trained on tokens, and a crow solving a novel puzzle are not commensurable quantities; the indexing is forced to be specific to the kind of system being measured, and cross-system comparison becomes a matter of convention rather than measurement. We take the denominator to be physical energy. Energy is defined identically across substrates, is measurable in joules without indexing dispute, and admits a clean decomposition into power and time when duration is operatively relevant. Choosing energy as denominator is not a cosmetic substitution. It produces a definition under which the deployed artifact and the process that produced it become distinct objects with potentially very different intelligence values, and under which inference-time computation is analysable as either update-energy or performance-energy depending on whether its products are incorporated into the behavioural state. Both consequences are absent from frameworks that index efficiency by experience or data, and both are developed formally below.

A clarification on the scope of substrate-neutrality. Substrate-neutrality is a property of the denominator. Energy is identically counted across biological, mechanical, and computational systems, requiring no substrate-specific indexing convention. The numerator — cost — is task-relative: a cost specification chooses what counts as success on a given task, and that choice is conventional. But once chosen, the specification applies identically to any system performing the task, so within a fixed cost specification cross-substrate comparison is well-defined. Energy supplies the denominator that survives the cross-substrate move; cost specification supplies a numerator that survives it for any fixed task. Frameworks that index efficiency by experience or sample count fail at the first step, because experience itself is not substrate-neutrally countable, so even within a fixed cost specification cross-substrate comparison is not licensed.

1.1 Exogeneity

A second structural advantage of the energy denominator is that energy is exogenous to the learning process in a way the alternatives are not. Priors, experience, sample counts, and curriculum interactions are all themselves outputs of prior learning: a system's prior at time t is a function of the learning that has occurred up to t, and a system's experience is the trajectory through which that learning was acquired. Frameworks that index learning efficiency by these quantities therefore measure efficiency relative to a reference state that is itself an output of the process being measured. This is a structural circularity, not a fatal one, but it creates well-known problems for cross-system comparison and for measuring learning trajectories where the reference state is still being assembled.

Energy does not have this property. Joules consumed are drawn from an external thermodynamic environment; the learning process is one of the things they get spent on, not one of the things that produces them. The denominator stands outside the dynamics it is measuring, in the way an external clock stands outside the events it is timing. This is the place where the energy choice does philosophical work that substrate-neutrality and measurability do not: it breaks the circularity that experience-indexed and prior-indexed frameworks inherit by construction.

A reviewer might object that the *quantity* of energy required for a given operation is itself endogenous to the learning state — a more competent system may extract more cost reduction per joule, so joules-spent are partly determined by where the system already is. This is correct, and it is a different point from the one being made. Energy is exogenous as a *resource type*: joules are drawn from outside the learning dynamics regardless of system state. The quantity required for a particular update is endogenous to the trajectory, just as the time taken to traverse a distance is endogenous to the traveller. This does not collapse the distinction between energy and experience as denominators. Experience-indexed frameworks have the stronger problem that the reference state itself — the priors against which efficiency is measured — is an output of the process being measured. Energy-indexed frameworks have only the weaker situation that the quantity of the (exogenous) resource required is sensitive to system state. The first is a circularity in the definition; the second is the ordinary state-dependence that any rate measurement has.

2 Scope and Specification

Intelligence values in this framework are defined relative to a *specification* consisting of:

1. a system boundary separating S from its producing process ρ (Section 6);
2. a task distribution $\mathcal{D}$ (Section 10);
3. a cost functional C with specified components and weights (Section 8);
4. (where amortised deliberation is evaluated) a fixed evaluation computation budget k_{eval} (Section 11.4).

Substrate-neutrality is a property of the denominator (energy is identically counted across substrates) and of within-specification comparison (a specified cost functional applies identically to whichever system performs the task). It is not a property of intelligence values in some absolute sense; cross-specification comparisons are not licensed by the framework.

3 Related Work

Four prior frameworks share territory with the present account.

Chollet's *On the Measure of Intelligence* (2019) defines intelligence as skill-acquisition efficiency relative to specified priors and experience, and explicitly distinguishes current skill from the rate at which skill is acquired. The level/slope cut is shared. Two structural differences with the present framework are worth noting. First, Chollet's denominator is experience, indexed by developer-specified task generators; this provides workable within-benchmark comparison but not substrate-neutral cross-system comparison, since experience is not substrate-neutrally countable. Second, and more deeply, Chollet's framework conditions on priors and experience that are themselves outputs of prior learning — a system's prior at time t is a function of learning up to t, and the system's experience is the trajectory along which that learning occurred. Skill-acquisition efficiency is therefore measured relative to a reference state that is itself an output of the process being measured. This is a structural circularity that energy-indexed efficiency does not have, since joules consumed are drawn from outside the learning dynamics rather than produced by them. Chollet's framework also offers no treatment of inference-time computation as either update or non-update, because the experience denominator does not licence the distinction.

The Legg–Hutter framework (2007) defines universal intelligence as a Solomonoff-weighted expectation of reward over computable environments. The expectation-over-tasks structure is shared. Legg–Hutter has no energetic cost structure: it is an idealised performance measure, not an efficiency measure, and the question of substrate or update-energy does not arise.

Friston-style free-energy accounts share substrate-neutral aspirations and operate in physical units. They specify a thermodynamic imperative — systems minimise variational free energy — rather than an efficiency metric over learnable cost. The free-energy principle answers a different question (why systems behave as they do) from the one this framework asks (how efficiently do they convert energy into cost reduction).

Bennett–Landauer-adjacent work on the thermodynamics of computation establishes lower bounds on the energy cost of computational operations and has long studied the relationship between energy and information. This bears on the floor of what is computationally possible, not on the cost-reduction-per-joule structure of learning. The thermodynamic floor is a constraint on the framework's denominator, not a competitor framework.

The present framework's distinguishing moves relative to these are: (i) energy as denominator with the substrate-neutral comparison this enables, (ii) the artifact/producing-process cleavage (Examples 11.1–11.2), and (iii) the amortised-deliberation analysis (Section 11.4), which has no obvious counterpart in experience-indexed or expectation-over-environments frameworks.

4 Generalisation Hierarchies

A prominent tradition in psychometrics organises cognitive abilities into a hierarchy of generality: narrow abilities at the base, broad factors in the middle, and a single general factor g at the apex. The Cattell–Horn–Carroll (CHC) model is the most developed version. In the intelligence-testing literature and in several recent AI-alignment proposals, this hierarchy is treated not merely as an empirical regularity among human test scores but as a structural property of intelligence itself—so that increasingly general competence is taken to *define* increasingly high intelligence. This section argues that the hierarchical picture, while empirically informative, should not be embedded in a formal definition.

4.1 What the hierarchical picture rests on

Three bodies of evidence motivate the staged view.

First, the psychometric evidence. Factor analyses of large batteries of cognitive tests consistently recover a hierarchical structure: specific tasks load on narrow factors, narrow factors load on broad factors (fluid reasoning, crystallised knowledge, processing speed, etc.), and broad factors correlate positively, yielding g. The hierarchy is robust across populations and test batteries, and g predicts a wide range of life outcomes.

Second, expertise research. Novices in a domain begin with narrow, context-bound skill; with practice, they acquire broader schemata that transfer across problems within the domain; with further practice some individuals generalise across related domains. The trajectory is local $\to$ broad, and the transition is reliably energy-intensive.

Third, developmental psychology. Infant cognition proceeds from reflexes through sensorimotor coordination to symbolic reasoning, a sequence that unfolds in a broadly consistent order across cultures. The progression is again from narrow to general.

4.2 Trajectories that do not match the climb

If staged generalisation were a structural property of intelligence, every efficient learning trajectory would ascend the same ladder: narrow first, broad later, extreme generality last. Several classes of system do not follow this path.

Example 4.1 (Breadth-first trajectories)**.** Large language models are pre-trained on broad, heterogeneous corpora—spanning languages, domains, and registers—before any task-specific fine-tuning. They acquire wide-ranging competence first and narrow expertise second, the reverse of the psychometric progression. Diffusion models similarly begin with a coarse, global structure (low-frequency content) and progressively refine local detail, achieving broad distributional coverage before fine-grained fidelity. These are not pathological cases; they are among the most effective learning procedures currently known.

Example 4.2 (Early broad competence in biological systems)**.** Infant cognition also presents cases that do not fit the narrow-first template. Neonates show broad statistical-learning abilities—tracking transitional probabilities across syllables, visual sequences, and tonal patterns—before any domain-specific expertise is in place. The starting point is a domain-general mechanism, not a narrow one.

The existence of efficient breadth-first and coarse-to-fine trajectories shows that the local → broad → extreme ordering is one empirically common pathway, not a necessary feature of cost-reducing systems.

4.3 Implications for formalisation

The evidence reviewed in Section 4.1 leaves little doubt that local-to-global generalisation is a common trajectory—perhaps the modal one—among biological learners operating under the energy and data constraints typical of terrestrial development. But “common” is not “necessary.” The counterexamples in Section 4.2 show that highly efficient cost reduction can proceed along quite different paths: breadth-first pre-training, coarse-to-fine refinement, or early domain-general statistical learning. What the current empirical picture supports is that staged generalisation is a frequently observed property of how systems *exhibit* intelligence under particular resource regimes, not a structural prerequisite for intelligence itself. Elevating it to a definitional requirement would mistake a prevalent trajectory for a universal law, and would exclude—by fiat rather than by evidence—any system whose most efficient route to low cost happens not to climb the narrow-to-broad ladder.

The present framework therefore treats generalisation breadth as a consequence of the task distribution $\mathcal{D}$, not as an axiom. A distribution that places weight on diverse, far-transfer tasks will, in practice, reward systems capable of broad generalisation; one concentrated on a single domain need not. Whether local-to-global ordering emerges as the efficient path under a given $\mathcal{D}$ is an empirical question the framework can investigate—it is not a constraint the framework imposes.

Remark 1 (Generalisation as empirical property)**.** Embedding staged generalisation in the definition of intelligence would:

(i) exclude systems whose most efficient trajectory is breadth-first or coarse-to-fine;

(ii) conflate the *order* in which generality is acquired with the *rate* at which cost is reduced; and

(iii) import an anthropocentric developmental sequence into a framework intended to be substrate-neutral.

The present framework avoids this by defining intelligence as cost reduction per unit energy across a task distribution $\mathcal{D}$, without prescribing the order in which tasks of different generality are mastered. If a system reduces cost most efficiently by starting broad, the definition registers that efficiency; it does not penalise the trajectory for failing to climb a predetermined ladder.

The task distribution $\mathcal{D}$ can, of course, be chosen to emphasise breadth, and a system that reduces cost across a wider distribution will, other things equal, score higher. But this is a property of the specification (Section 2), not of the definition. Whether generality correlates with intelligence is an empirical question that the framework can express; it is not an axiom the framework assumes.

5 Heuristic Definition

Definition 1 (Heuristic definition of intelligence)**.** Intelligence is the capacity of a system to reduce the cost of performing tasks through the expenditure of energy.

A task may be a physical action, a cognitive operation, or any goal-directed interaction with an environment. Its cost may include time, effort, error, risk, or any other resource consumed in producing the relevant outcome. Energy consumption provides a common physical measure for comparing the resources expended by different systems in acquiring or improving task competence. In contrast to denominators such as “experience” or “data”, energy is a substrate-neutral physical unit defined uniformly across biological, mechanical, and computational systems. It therefore permits cross-agent comparison without recourse to incommensurable indexing.

On this view, learning to ride a bicycle consists in reducing the cost per successful metre of cycling through energy expenditure: less conscious coordination, fewer falls, less muscular tension, and smoother control are achieved as the organism consumes metabolic energy over practice. Learning calculus similarly consists in reducing the cost of solving calculus-related tasks: derivatives, integrals, and limits become faster, less error-prone, and less effortful as energy is expended on study, practice, and cognitive updating.

Intelligence is not merely having a low cost at a given moment. It is the capacity to make that cost fall for a given amount of energy consumed. A system is more intelligent, in this sense, insofar as it reaches lower task cost, or a given level of competence, with less energy expenditure.

This separates three related notions.

Competence How costly it currently is for a system to perform a task.

Learning The process by which that cost decreases as energy is expended.

Intelligence The capacity or rate at which a system can produce such cost reductions per unit of energy consumed across tasks.

A static lookup table of optimal actions can therefore be highly competent, because it may assign very low cost to a task. But by itself it is not intelligent in this dynamic sense, since it has no mechanism for adding, revising, or improving its entries through energy-consuming update. If the lookup table is embedded within a wider system capable of updating it through energy expenditure, then the wider system can be evaluated for intelligence.

Remark 2. A note on scope. The framework is a definition, not a benchmark, a measurement protocol, or a universal prior over tasks. Its primary outputs are conceptual cuts — between competence and intelligence, between artifact and producing process, and between movement along a fixed-state performance frontier and improvement of the frontier itself. Empirical variants suitable for measurement are recommended in Appendix B, but the central claim of the paper is that these cuts are the right ones to make, not that any particular operationalisation of them is final.

6 System Individuation

Let S be a system and let ρ denote a producing process that brings S into existence. Such a process may include parents, evolutionary history, prior model versions, research effort, or other antecedent causal machinery. The energy consumed by ρ is not part of S's energy budget. How a system came into being is distinct from what it does once it exists.

Convention 1 (System boundary)**.** The boundary between S and ρ is fixed by convention, but by a principled convention: S begins where its own energy expenditure begins to update its own behavioural state.

For a human, this excludes parental and evolutionary energy. For an AI system, this excludes the energy consumed by researchers and prior model lineage, but includes the training run that sets its weights, including subsequent fine-tuning that continues to update those weights.

Cases at the boundary, such as distillation into a smaller model, weight merging, and gradual retraining, admit reasonable disagreement. They should be settled by explicit convention applied consistently within a given evaluation.

7 Total Energy

The energy attributed to S is its total energy consumption, not only the portion identifiable as "learning energy". A system that consumes large amounts of energy on activity unrelated to updating its behavioural state will, by definition, exhibit a lower cost-reduction rate per unit of energy and so register as less intelligent.

This is intended behaviour. Thermodynamic and metabolic efficiency are part of what is being measured. There is no separate accounting for "productive" versus "wasted" energy. An organism or model that burns substantial energy without extracting corresponding cost reduction is, on this view, less intelligent than one that achieves the same reduction more frugally.

7.1 Power and Time

Where time is a binding constraint on the evaluation, energy may be expressed as the integral of power consumption over time:

$$E = \int_{t_0}^{t_1} P(t)\, \mathrm{d}t.$$

This admits two complementary views of intelligence. With time fixed, intelligence may be evaluated against power consumption, capturing how much cost reduction a system produces per unit of power within a bounded interval. With time unconstrained, intelligence may be evaluated against total energy, capturing cumulative efficiency irrespective of duration. The choice depends on whether time is an operative constraint on the task in question; the underlying cost-reduction-per-resource form is unchanged.

Remark 3 (The anthropocentric special case)**.** Under constant power P, energy is $E = P\Delta t$, so time becomes a positive affine rescaling of energy. This is the regime in which the folk comparison — “they learned it in half the time” — coincides with the energy-indexed definition. Human cerebral power draw is approximately constant at roughly 20 watts across individuals and across waking hours. The familiar time-indexed intelligence comparison therefore agrees with the energy-indexed ranking for humans, not because time is a good general denominator, but because the constant-power assumption happens to hold. The agreement breaks the moment systems with different power consumption are compared: a GPU cluster drawing megawatts and a human brain drawing 20 watts cannot be ranked by time-to-learn, because the same duration corresponds to energy expenditures differing by five orders of magnitude. What looks like a natural measure of intelligence — how fast did you learn? — is an anthropocentric special case of the energy measure, valid precisely where metabolic power is approximately constant.

8 The Cost Functional

Let $\mathcal{T}$ be a class of tasks, and let $\tau \in \mathcal{T}$ denote a particular task. Let S be a system whose behaviour on τ is described by a policy, procedure, or behavioural state π_E, where $E \geq 0$ indexes accumulated energy consumption. Energy consumption may correspond to metabolic energy, electrical energy, computational energy, mechanical energy, or any other physical energy expended by the system in updating its behavioural state.

Definition 2 (Expected task cost)**.** The expected cost incurred by system S, in behavioural state π_E, when performing task τ, is a functional

$$C(\pi_E, \tau) \in \mathbb{R}_{\geq 0}.$$

The cost functional may aggregate multiple dimensions. For example,

$$C(\pi_E, \tau) = \mathbb{E}\left[\alpha T + \beta F + \gamma R + \delta L\right],$$

where T is time, F is effort or exertion, R is risk, L is error or loss, and

$$\alpha, \beta, \gamma, \delta \in \mathbb{R}_{\geq 0}$$

determine their relative importance.

The weights $\alpha, \beta, \gamma, \delta$ are placeholders. Real cost functions in real domains are not generally tractable to write down in closed form, and their weights are domain-dependent and partly empirical. The schema asserts that cost is multidimensional and that its components trade off; it does not prescribe the trade-off.

Energy E is not included as one cost term among others here. Rather, it is the independent resource relative to which improvement is measured. The question is not merely how costly the task currently is, but how much that cost falls per unit of energy consumed.

8.1 Cost-relativity

Intelligence in this framework is defined relative to a specified cost functional, in the same sense that velocity is defined relative to a specified reference frame. Two systems may rank in opposite orders under different cost specifications, just as two objects may have different relative velocities in different frames; this is not a defect of the underlying concept but a consequence of its relational structure. Affine transformations of cost (unit changes, baseline offsets) preserve rankings within a specification (Proposition A.8). Nonlinear monotone transformations such as squaring or log-loss conversion (Proposition A.9) do not preserve rankings, because they encode substantively different cost specifications — different attitudes toward how large errors trade off against small ones — rather than different units of the same specification. The framework therefore demands an explicit cost specification as input to any evaluation, just as it demands an explicit system boundary and task distribution.

8.2 Cost transformations as world-changes

The framework's behaviour under cost transformations admits a sharper interpretation than mere ranking-relativity. Affine transformations of cost — rescalings and offsets — correspond to changes in the units in which cost is measured. They preserve the ranking of intelligence values because they preserve the structure of the cost landscape: the agent is operating in the same world, described in different units. Nonlinear monotone transformations are different in kind. They change the trade-off between large and small errors, which is to say they change how the environment penalises mistakes of different magnitudes. These are not unit changes. They are changes in the error-tolerance structure of the environment — changes in what the world is like.

Example 8.1 (Affine cost transformations preserve rankings)**.** Measuring task cost in seconds versus milliseconds is multiplication by 1000 — a positive affine rescaling. If system A reduces expected time-cost per joule faster than system B in seconds, it does so in milliseconds too. Adding a fixed baseline overhead to every task — say 5 units of cost regardless of performance — shifts the cost surface uniformly. The derivative $-\mathrm{d}C/\mathrm{d}E$ is unchanged because the constant disappears under differentiation. In both cases, the world has not changed; only its description has. The anthropocentric special case — "theylearned it in half the time" as an affine rescaling of energy under constant metabolic power — is discussed in Section 6.1.

The following examples illustrate the converse: nonlinear monotone transformations that do not merely relabel cost but reshape the cost landscape in ways that change which learning trajectories are efficient, first mathematically and then in terms of the kind of world each transformation describes.

Example 8.2 (Squared cost — the minefield regime)**.** Let two systems have the following cost trajectories on a single task over an energy interval $[0, \Delta E]$:

$$C_A(E) = 10 - 2E, \qquad -\frac{\mathrm{d}C_A}{\mathrm{d}E} = 2,$$

$$C_B(E) = 3 - E, \qquad -\frac{\mathrm{d}C_B}{\mathrm{d}E} = 1.$$

Under linear cost, $\mathrm{Int}_A = 2 > 1 = \mathrm{Int}_B$: system A ranks above B. Now apply the transformation $C \mapsto C^2$. The transformed learning rate is

$$-\frac{\mathrm{d}(C^2)}{\mathrm{d}E} = 2C\left(-\frac{\mathrm{d}C}{\mathrm{d}E}\right).$$

At $E = 0$: for A, this is $2(10)(2) = 40$; for B, this is $2(3)(1) = 6$. System A still leads. But the factor $2C$ means the transformed rate is proportional to current cost. As A's cost falls toward B's, its advantage shrinks; and if we compare systems with the values from Proposition A.9 — $C_1 = 1$ with rate 2 versus $C_2 = 10$ with rate 1 — the ranking reverses outright: A scores 4 while B scores 20 under squared cost.
Squaring cost describes a world where large errors are catastrophically more expensive than small ones — a minefield. An exploratory learner that accepts large early errors to gather information has those errors amplified quadratically. A cautious learner that keeps cost low throughout avoids the amplification zone. The ranking reversal is the framework correctly recording that exploratory learning is efficient in a forgiving world and inefficient in a minefield.

Example 8.3 (Log cost — the diminishing-penalty regime)**.** Apply the transformation $C \mapsto \log(1 + C)$. The transformed learning rate is

$$-\frac{\mathrm{d}\log(1+C)}{\mathrm{d}E} = \frac{1}{1+C}\left(-\frac{\mathrm{d}C}{\mathrm{d}E}\right).$$

The factor $1/(1+C)$ is large when C is small and small when C is large. A system operating at high cost — making large, coarse improvements — has its rate of cost reduction discounted. A system operating near zero cost — polishing already-good performance — has its rate amplified. Consider two systems at $E = 0$: system A has $C_A = 100$ and $-\mathrm{d}C_A/\mathrm{d}E = 50$; system B has $C_B = 1$ and $-\mathrm{d}C_B/\mathrm{d}E = 1$. Under linear cost, A dominates $(50 > 1)$. Under log cost, A's transformed rate is $50/101 \approx 0.50$ while B's is $1/2 = 0.50$ — they are tied, despite a factor-of-fifty difference in raw rate. Adjust the numbers slightly and B leads.
Log cost describes a world of diminishing penalties — one where the difference between terrible and bad matters much less than the difference between good and excellent. Medical diagnostics after initial triage is such a world: once you have identified the correct disease category (large cost reduction, from "no idea" to "roughly right"), the remaining value is in refining the diagnosis (small cost reduction, from "roughly right" to "precisely right"), and that refinement is where error matters most. A system that is good at refinement but slow at initial triage can rank above a system that triages quickly but plateaus at moderate accuracy, because the log cost specification says the refinement work is where the real cost lives.

Example 8.4 (Threshold cost — the pass/fail regime)**.** Define threshold cost as $C^{\text{thresh}}(\pi_E, \tau) = \mathbf{1}[C(\pi_E, \tau) > c^*]$ for a fixed threshold c^*. Under continuous cost, a system that steadily reduces cost from 10 to 3 over an energy interval has positive intelligence throughout — the derivative is negative, cost is falling. Under threshold cost with $c^* = 1$, that same trajectory registers $C^{\text{thresh}} = 1$ throughout the interval (the system is above threshold from start to finish), so

$$-\frac{\mathrm{d}C^{\text{thresh}}}{\mathrm{d}E} = 0.$$

All the above-threshold improvement is invisible. A different system that reduces cost from 1.5 to 0.5 — a much smaller absolute reduction — crosses the threshold, producing a

discrete jump from $C^{\text{thresh}} = 1$ to $C^{\text{thresh}} = 0$. That system has positive intelligence at the crossing point; the first system has zero.
Threshold cost describes a pass/fail world — a licensing exam, a minimum-viability product gate, a survival threshold in a hostile environment. In such a world, partial progress below the threshold does not count. The framework correctly records that a system which crosses the threshold per joule of update-energy is more intelligent, in this world, than one which makes larger absolute improvements that never cross. This is not a pathology; it is the correct behaviour for a world that does not reward partial credit.

Under this reading, the failure of nonlinear invariance is not a defect of the framework but a faithful tracking of the fact that intelligence is relational to the cost landscape an agent inhabits. The same system, with the same internal dynamics, can rank differently across these regimes, and the framework correctly registers this. Intelligence rankings reordering under nonlinear cost transformation is not the framework misbehaving; it is the framework recording that the environment's error-tolerance structure has changed in a way that genuinely matters for which update trajectories reduce cost efficiently.

This is the sense in which intelligence in the present framework is irreducibly relational to a world. Affine invariance corresponds to "the world is the same, described differently." Nonlinear non-invariance corresponds to "the world is different, and the rate at which a given learning trajectory reduces cost has changed with it." The framework is not making a claim that intelligence values are arbitrary or convention-bound; it is making the stronger claim that intelligence is a property of an agent-world pairing, and that changes to either side of the pairing legitimately move the rankings.

9 Core Definitions

9.1 Competence

Definition 3 (Competence)**.** The competence of system S on task τ after consuming E units of energy is inversely related to its current cost:

$$\mathrm{Comp}_S(\tau, E) := -C(\pi_E, \tau).$$

Equivalently, competence may be represented by any monotone decreasing transformation of $C(\pi_E, \tau)$. Lower cost means greater competence.

Ordinal character of competence. Competence is primarily an ordinal notion. If system S_1 has lower expected cost than S_2 on task τ, then S_1 is more competent than S_2 on τ. The numerical representation $\mathrm{Comp}_S(\tau, E) = -C(\pi_E, \tau)$ is not unique; it is a convenient cardinalisation of an underlying order.

Proposition 9.1 (Competence rankings are invariant under monotone reparameterisation)**.** *Let competence be represented by* $\mathrm{Comp}_S(\tau, E) = g(C(\pi_E, \tau))$*, where* $g : \mathbb{R}_{\geq 0} \to \mathbb{R}$ *is strictly decreasing. For any two systems* S_1, S_2*,*

$$C(\pi_E^1, \tau) < C(\pi_E^2, \tau) \qquad \textit{if and only if} \qquad \mathrm{Comp}_{S_1}(\tau, E) > \mathrm{Comp}_{S_2}(\tau, E).$$

Hence any strictly decreasing reparameterisation of cost preserves competence rankings on *a fixed task under a fixed cost specification.* (Proof in Appendix A.1.)

This invariance is stronger than the corresponding invariance for intelligence. Because competence is a level, any monotone reparameterisation preserves its order. Because intelligence is a slope — a derivative of cost with respect to energy — nonlinear monotone transformations can change rankings by changing the marginal weight assigned to reductions at different cost levels (Proposition A.9; Section 7.2). Competence is ordinally robust; intelligence is cardinally cost-sensitive.

Definition 4 (Competence relative to a task distribution)**.** The competence of system S relative to a task distribution $\mathcal{D}$ after consuming update-energy E is inversely related to expected current cost:

$$\mathrm{Comp}_S(\mathcal{D}, E) := -\mathbb{E}_{\tau\sim\mathcal{D}}\left[C(\pi_E, \tau)\right].$$

Equivalently, one may represent distributional competence by any strictly decreasing transformation of $\mathbb{E}_{\tau\sim\mathcal{D}}[C(\pi_E, \tau)]$.

This distributional quantity is only meaningful relative to the same $\mathcal{D}$ and the same cost functional. A system may be highly competent relative to $\mathcal{D}_1$ and incompetent relative to $\mathcal{D}_2$. Competence, like intelligence, is not specification-free.

Competence and deployment computation. Where deployment-time computation matters, competence should be indexed by an evaluation budget k:

$$\mathrm{Comp}_S(\tau, E; k) := -C_{\mathrm{perf}}(\pi_E, k, \tau).$$

This measures how competent the system is when allowed k units of inference-time computation while holding the behavioural state fixed. Increasing k may increase competence without increasing intelligence, because it may reduce performance cost without changing π_E. Competence can therefore be budget-relative in a way intelligence is not: intelligence concerns the change of the behavioural state with update-energy, whereas deployment competence concerns the cost achieved by a given state under a specified performance budget. This notation is used extensively in Section 11.

Example 9.1 (Static high competence)**.** A frozen chess engine may have very high competence on chess tasks because its expected cost of move selection is low. If its behavioural state does not update during the deployment interval, this high competence does not imply intelligence over that interval. It is a high level, not a positive slope.

Example 9.2 (Competence depends on allowed computation)**.** A reasoning model may answer poorly under a small inference budget and well under a large one. At fixed behavioural state π_E, this is a difference in deployment competence:

$$C_{\mathrm{perf}}(\pi_E, k_{\mathrm{hi}}, \tau) < C_{\mathrm{perf}}(\pi_E, k_{\mathrm{lo}}, \tau).$$

The system is more competent at k_{hi} than at k_{lo}, but since π_E has not changed, this is not intelligence. It is movement along a fixed-state performance frontier.

9.2 Learning

Definition 5 (Learning over an energy interval)**.** System S learns task τ over an interval of energy consumption $[E_0, E_1]$ when its expected task cost decreases over that interval:

$$C(\pi_{E_1}, \tau) < C(\pi_{E_0}, \tau).$$

The net learning over the interval is

$$L_S(\tau; E_0, E_1) := C(\pi_{E_0}, \tau) - C(\pi_{E_1}, \tau).$$

Thus $L_S > 0$ is positive learning, $L_S < 0$ is negative learning (forgetting, maladaptation, or interference relative to the task and cost specification), and $L_S = 0$ is no net learning relative to τ. The interval definition measures net learning between endpoints, not the total amount of state change or exploratory motion inside the interval; a system may undergo large behavioural changes and still have $L_S = 0$ if its endpoint cost equals its initial cost.

In differential form, local learning occurs at energy level E when

$$\frac{\mathrm{d}}{\mathrm{d}E} C(\pi_E, \tau) < 0.$$

The local learning rate is

$$\lambda_S(\tau, E) := -\frac{\mathrm{d}}{\mathrm{d}E} C(\pi_E, \tau).$$

Positive λ_S indicates instantaneous cost reduction per unit of update-energy; negative λ_S indicates instantaneous cost increase.

Learning is not state change. An energy-consuming change in behavioural state counts as learning only if it lowers expected cost under the relevant task and cost specification. Random drift, unhelpful plasticity, destructive fine-tuning, and irrelevant memorisation may change π_E without producing positive learning on the task of interest. The update operator introduced in Section 11 formalises this: an update $U(\pi_E, e_E) = \pi_{E+\Delta E}$ is learning on τ only if $C(\pi_{E+\Delta E}, \tau) < C(\pi_E, \tau)$.

Learning is task-relative. The same update can reduce cost on one task while increasing cost on another. Learning is therefore task-relative before it is distribution-relative; it becomes distribution-relative only after a task distribution $\mathcal{D}$ is specified.

Proposition 9.2 (Learning on one task need not imply learning on another)**.** *There exist tasks τ_1, τ_2 and an update interval $[E_0, E_1]$ such that*

$$C(\pi_{E_1}, \tau_1) < C(\pi_{E_0}, \tau_1) \qquad \textit{while} \qquad C(\pi_{E_1}, \tau_2) > C(\pi_{E_0}, \tau_2).$$

Thus an update can constitute learning relative to one task and negative learning relative to another. (Proof in Appendix A.2.)

Proposition 9.3 (The sign of learning is ordinally invariant; the magnitude is not)**.** *Let $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be strictly increasing. Then*

$$C(\pi_{E_1}, \tau) < C(\pi_{E_0}, \tau) \qquad \textit{if and only if} \qquad f(C(\pi_{E_1}, \tau)) < f(C(\pi_{E_0}, \tau)).$$

Therefore whether positive learning occurred over an interval is invariant under strictly *increasing transformations of cost. However, the magnitude $L_S(\tau; E_0, E_1) = C(\pi_{E_0}, \tau) - C(\pi_{E_1}, \tau)$ and the rate obtained by dividing it by energy are not invariant under nonlinear transformations.* (Proof in Appendix A.2.)

This gives a three-level invariance structure across the core definitions. Competence rankings are invariant under all strictly monotone transformations of cost (Proposition 9.1). The sign of learning — whether cost fell, rose, or was unchanged — is likewise invariant under all strictly monotone transformations. But the magnitude of learning and the rate of intelligence are invariant only under positive affine rescaling (Proposition A.8), not under arbitrary nonlinear monotone transforms (Proposition A.9). The invariance weakens as the concept moves from level to sign to rate.

Example 9.3 (Forgetting)**.** A model fine-tuned on a narrow distribution may reduce cost on the fine-tuning tasks while increasing cost on previously mastered tasks. Relative to the fine-tuning distribution, $L_S > 0$: the system learns. Relative to the original broad distribution, $L_S < 0$: the system exhibits negative learning. This is catastrophic forgetting expressed in the framework's terms.

Example 9.4 (Exploration without net learning)**.** A reinforcement learner may consume energy exploring many behaviours, update its state repeatedly, and nevertheless return to a policy with the same expected cost at E_1 as at E_0. It has changed state — $\pi_{E_1} \neq \pi_{E_0}$ — but $L_S(\tau; E_0, E_1) = 0$. State change without cost reduction is not learning in this framework.

9.3 Intelligence

Definition 6 (Intelligence relative to a task distribution)**.** Let $\mathcal{D}$ be a distribution over tasks. The local intelligence of system S, relative to $\mathcal{D}$, is the expected rate at which it reduces task cost per unit of update-energy:

$$\mathrm{Int}_S(\mathcal{D}, E) := \mathbb{E}_{\tau \sim \mathcal{D}} \left[-\frac{\mathrm{d}}{\mathrm{d}E} C(\pi_E, \tau) \right].$$

Equivalently,

$$\mathrm{Int}_S(\mathcal{D}, E) = \mathbb{E}_{\tau \sim \mathcal{D}} [\lambda_S(\tau, E)].$$

Intelligence is therefore distributional learning rate: the expectation, under $\mathcal{D}$, of the task-relative learning rates defined above.

For discrete energy intervals, the corresponding finite-difference form is

$$\mathrm{Int}_S(\mathcal{D}; E_0, E_1) := \frac{\mathbb{E}_{\tau \sim \mathcal{D}} [C(\pi_{E_0}, \tau) - C(\pi_{E_1}, \tau)]}{E_1 - E_0}, \qquad E_1 > E_0.$$

This is the empirical form of the definition: it measures net expected cost reduction per unit of update-energy over the interval.

Intelligence is signed and distribution-relative. Positive values indicate expected cost reduction per unit update-energy. Zero indicates no net expected improvement relative to $\mathcal{D}$. Negative values indicate that energy-consuming updates increase expected task cost relative to $\mathcal{D}$, as in forgetting, destructive fine-tuning, or maladaptive learning. An update can be intelligent relative to one distribution and negatively intelligent relative to another. Positive intelligence under $\mathcal{D}$ does not require positive learning on every task in the support of $\mathcal{D}$; it requires positive expected improvement under the task weighting supplied by $\mathcal{D}$.

Raw intelligence and its sensitivities. The quantity above is *raw* local intelligence: absolute expected cost reduction per joule. Because it is a raw derivative, it is sensitive to the remaining reducible cost — a system starting from worse competence may exhibit larger raw intelligence than a better system with less room left to improve. It is also sensitive to the cardinal structure of the cost functional and to the chosen task distribution. These sensitivities are not defects of the definition; they specify what the raw quantity measures. Where the intended question is how efficiently a system closes its remaining gap to an asymptotic floor, a proportional variant is preferable:

$$\widetilde{\mathrm{Int}}_S(\mathcal{D}, E) = \mathbb{E}_{\tau\sim\mathcal{D}}\left[-\frac{1}{C_S(E,\tau) - C_S^{\infty}(\tau)}\frac{\mathrm{d}C_S(E,\tau)}{\mathrm{d}E}\right],$$

when an interpretable cost floor $C_S^{\infty}(\tau)$ is available. Appendix B records this and other empirical variants that remove or control for particular sensitivities.

Intelligence depends on the cardinal cost specification, continuing the invariance hierarchy established in the preceding subsections. Competence rankings are preserved by arbitrary monotone reparameterisations of cost (Proposition 9.1); the sign of learning over an interval is likewise preserved (Proposition 9.3); but intelligence rankings are not generally preserved by nonlinear monotone transformations, because rates depend on the marginal geometry of cost. This is why the cost functional is part of the evaluation specification rather than a harmless notational choice.

Proposition 9.4 (Interval intelligence composes as an energy-weighted average)**.** *For* $E_0 < E_1 < E_2$,

$$\mathrm{Int}_S(\mathcal{D}; E_0, E_2) = \frac{(E_1 - E_0)\,\mathrm{Int}_S(\mathcal{D}; E_0, E_1) + (E_2 - E_1)\,\mathrm{Int}_S(\mathcal{D}; E_1, E_2)}{E_2 - E_0}.$$

Thus intelligence over a long interval is not additive. Net learning is additive (Proposition A.6); intelligence is the rate obtained after normalising by energy, and rates average rather than sum. (Proof in Appendix A.3.)

An improvement-threshold form of intelligence, which asks how many joules are needed to purchase a specified cost reduction ΔC, is developed in Appendix A.3. It avoids the pathology of target-competence thresholds, under which systems already below a target cost receive infinite or undefined intelligence values.

Zero intelligence is specification-relative. A frozen deployed model has zero intelligence under a system boundary that excludes the training pipeline. But if the boundary is expanded to include a memory store, fine-tuning loop, retrieval-augmented update, or distillation pipeline, the larger system may have positive intelligence. The claim that a frozen artifact has zero intelligence is not the claim that the broader process that produces or maintains it has zero intelligence; it is a claim relative to the specified boundary and update interval.

Example 9.5 (Separating competence, intelligence, and transfer)**.** A frozen expert policy may have high competence and zero intelligence over the deployment interval. A poor but rapidly improving learner may have low competence and positive intelligence. A fine-tuning run that improves one distribution while damaging another may have positive intelligence relative to the fine-tuning distribution and negative intelligence relative to the original broad distribution. These are not edge cases; they are the ordinary behaviour of the definitions once the level/slope/distribution distinctions are in view.

10 The Task Distribution

The framework is parameterised by a distribution $\mathcal{D}$ over tasks. In the limiting case, $\mathcal{D}$ is the space of all tasks expressible in the universe: a perfect world model. Restrictions of $\mathcal{D}$ to particular task families are admissible for practical evaluation but become incoherent as measures of intelligence below a certain breadth. When $\mathcal{D}$ is too narrow, the metric ceases to measure intelligence and instead measures memorisation or task-specific competence on a fixed domain.

Sufficient breadth of $\mathcal{D}$ is therefore a precondition for the metric to track what is intended by the term "intelligence" rather than something narrower. This is not merely a technical caveat: the breadth condition encodes the anti-memorisation pressure that distinguishes intelligence from domain-specific competence. A system that achieves perfect cost reduction on a narrow $\mathcal{D}$ by storing exact-match responses to seen tasks scores arbitrarily high on the raw metric and arbitrarily low on transfer. The breadth requirement is what prevents the framework from rewarding memorisation in the limit.

In the universe-$\mathcal{D}$ limit, a hypothetical lookup table containing optimal actions for every task, with perfect recall and zero recall cost, would be indistinguishable from perfect intelligence and perfect competence. The universe-$\mathcal{D}$ limit exhibits the saturation behaviour characteristic of relational concepts. Speed is undefined for a system with no remaining distance; a healing rate is undefined for a system with no remaining injury. Intelligence-as-cost-reduction is relational in the same way: at the universe-$\mathcal{D}$ limit there is no remaining cost to reduce, and the rate-of-reduction concept retires at its domain boundary. The lookup-table case is not a case where the framework gives wrong answers in normal use; it is the boundary at which the concept stops applying. The deeper question — what should be credited with the work that produced the perfect lookup table — is answered by the artifact/process distinction: the producing process that drove cost from its initial value to zero has high integrated intelligence, even though the resulting artifact has no remaining work to do. In any realistic setting no system possesses such a table, and the distinction remains operative.

The breadth requirement admits an operational test. If a system has high apparent intelligence on $\mathcal{D}$ and low transfer intelligence on a held-out distribution $\mathcal{D}_{\text{test}}$ (Appendix B; Proposition A.7), this is evidence that $\mathcal{D}$ was too narrow and the score reflected memorisation or domain-specific competence rather than intelligence in the intended sense. The $\mathcal{D}_{\text{train}} / \mathcal{D}_{\text{test}}$ split is the standard cross-validation pattern adapted to this metric, and it is what makes the breadth requirement a testable convention rather than a declared one. Practical evaluations should therefore report both within-$\mathcal{D}$ intelligence and transfer intelligence; agreement between them is evidence the breadth convention was met, and disagreement diagnoses the failure.

11 Updates, Inference, and Amortised Deliberation

11.1 Levels, slopes, and producing processes

Ordinary usage applies "intelligent" to at least three distinguishable things: (a) a system's current capability on tasks, (b) its capacity to improve that capability through further experience or training, and (c) the producing process that generated the system's current state. Biological cases rarely separate these — a competent human is typically also one whose state continues to update — and ordinary usage has not needed to distinguish them. Frozen AI systems separate them sharply: a fixed-weight deployed model has high (a), zero (b) during the deployment interval, and inherits its (a) from a (c) that may have very high values during training. The framework assigns (a) to *competence*, (b) to *intelligence over an update interval*, and (c) to the *intelligence of the producing process*. The frozen-model result that follows from this — $\mathrm{Int}_S = 0$ for a deployed copy with no parameter updates — is not a paradox or a counterintuitive discovery; it is what falls out once these three uses of the word are kept separate.

11.2 The Update Operator

To distinguish a merely static system from a dynamically intelligent one, assume that the system has an energy-consuming update operator

$$U : (\pi_E, e_E) \longmapsto \pi_{E+\Delta E},$$

where e_E is new input, interaction, observation, or training signal acquired through the expenditure of energy ΔE.

A system exhibits intelligence only insofar as this update process tends to reduce task cost per unit of energy consumed:

$$C\big(U(\pi_E, e_E), \tau\big) < C(\pi_E, \tau)$$

for relevant tasks τ, or in expectation across $\tau \sim \mathcal{D}$, with the reduction evaluated relative to the energy consumed in producing the update.

A static table of optimal actions may therefore have high competence, but without a nontrivial energy-consuming update operator it has zero learning rate and hence zero intelligence in this sense, with the universe-$\mathcal{D}$ limiting case noted above.

Example 11.1 (A frozen frontier reasoning model)**.** Let S be a particular deployed copy of a frontier reasoning model, for instance a fixed-weight copy of GPT-5.5 Thinking, with no parameter update and no persistent memory update during the interval under evaluation. The model name is incidental; the relevant assumption is that

$$\pi_E = \pi_{E'} \qquad \text{for all } E, E' \text{ in the deployment interval.}$$

Then, for every task τ,

$$C(\pi_E, \tau) = C(\pi_{E'}, \tau),$$

so

$$\lambda_S(\tau, E) = 0 \qquad \text{and hence} \qquad \mathrm{Int}_S(\mathcal{D}, E) = 0.$$

This is not paradoxical once the level/slope/producing-process distinction is in view (see "Levels, slopes, and producing processes" above): what is intelligent is the producing process, not the frozen artifact. This remains true even if the model has high competence or high expressed performance. For example, at a large inference-time computation budget k_{hi}, it may have very low performance cost

$$C_{\mathrm{perf}}(\pi_E, k_{\mathrm{hi}}, \tau) \ll C_{\mathrm{perf}}(\pi_E, k_{\mathrm{lo}}, \tau)$$

for some $k_{\mathrm{hi}} > k_{\mathrm{lo}}$. That inequality records deliberative gain along a fixed-state performance frontier. It does not record intelligence unless the behavioural state itself is updated.

Example 11.2 (The system that trains the model)**.** Let S_{train} be the training and post-training system that produces or modifies the deployed model. Its behavioural state includes the model parameters and any other state variables that determine future behaviour. If the training system consumes energy over an interval $[E_0, E_1]$ and produces a new behavioural state satisfying

$$\mathbb{E}_{\tau\sim\mathcal{D}}\left[C(\pi_{E_1}, \tau)\right] < \mathbb{E}_{\tau\sim\mathcal{D}}\left[C(\pi_{E_0}, \tau)\right],$$

then its interval intelligence is positive:

$$\text{Int}_{S_{\text{train}}}(\mathcal{D}; E_0, E_1) = \frac{\mathbb{E}_{\tau\sim\mathcal{D}}\left[C(\pi_{E_0}, \tau) - C(\pi_{E_1}, \tau)\right]}{E_1 - E_0} > 0.$$

Thus a deployed frozen model may have zero intelligence in the present sense, while the larger system that trains, fine-tunes, distils, or otherwise updates it may have positive intelligence. The distinction is not between biological and artificial systems; it is between fixed-state performance and energy-consuming state improvement.

11.3 Deliberation along a fixed-state frontier

Corollary 1 (Inference-time reasoning increases performance, not intelligence)**.** *Let π_E denote a system's behavioural state after update-energy E. Let $k \geq 0$ denote inference-time computation or performance-energy expended at deployment, such as extended deliberation, chain-of-thought, search, sampling, or tool use. Write*

$$C_{\text{perf}}(\pi_E, k, \tau)$$

for the performance cost incurred on task τ when the system expends k units of inference-time computation while holding π_E fixed.
It may be true that, for $k_2 > k_1$,

$$C_{\text{perf}}(\pi_E, k_2, \tau) < C_{\text{perf}}(\pi_E, k_1, \tau).$$

Such an inequality records improvement along a fixed-state performance frontier. It does not, by itself, record intelligence in the update sense, because the behavioural state has not changed. Intelligence concerns reduction of $C(\pi_E, \tau)$ as update-energy changes E, not merely reduction of performance cost as deployment computation k changes with π_E held fixed.
Thus inference-time reasoning can improve performance or expressed competence without constituting intelligence. What is intelligent, in this picture, is the energy-consuming update process that produced a behavioural state π_E whose fixed-state performance frontier improves with additional k.

11.4 Amortised Deliberation

The preceding corollary concerns inference-time reasoning that leaves the behavioural state fixed. This subsection considers a second case: the products of such reasoning may themselves become update signals. This matters because it turns fixed-state performance improvement into ordinary interval intelligence once the resulting signal is incorporated through the update operator.

The distinction is between movement along a fixed-state frontier and improvement of the frontier itself. The former is indexed by inference-time computation k. The latter is indexed by update-energy E.

Definition 7 (Fixed-state performance frontier)**.** For system S, task τ, and behavioural state π_E, the fixed-state performance frontier is the function

$$F_{S,E,\tau}(k) := C_{\text{perf}}(\pi_E, k, \tau), \qquad k \geq 0.$$

It records how performance cost varies with inference-time computation while π_E is held fixed.

Definition 8 (Deliberative gain)**.** For $0 \leq k_0 < k_1$, the deliberative gain of S on task τ at behavioural state π_E is

$$G_S(\tau, E; k_0, k_1) := C_{\text{perf}}(\pi_E, k_0, \tau) - C_{\text{perf}}(\pi_E, k_1, \tau).$$

Relative to a task distribution $\mathcal{D}$, define

$$G_S(\mathcal{D}, E; k_0, k_1) := \mathbb{E}_{\tau \sim \mathcal{D}} \left[G_S(\tau, E; k_0, k_1) \right].$$

Proposition 11.1 (Deliberative gain is not intelligence)**.** *A system may have positive deliberative gain while having zero intelligence over the same interval of update-energy.*

Proof. Let S be update-static over an interval, so that π_E is constant throughout that interval. Then $C(\pi_E, \tau)$ is constant as a function of E, and hence

$$\lambda_S(\tau, E) = -\frac{\mathrm{d}}{\mathrm{d}E} C(\pi_E, \tau) = 0.$$

Thus $\mathrm{Int}_S(\mathcal{D}, E) = 0$. Nevertheless, it may be true that for $k_1 > k_0$,

$$C_{\text{perf}}(\pi_E, k_1, \tau) < C_{\text{perf}}(\pi_E, k_0, \tau),$$

so $G_S(\tau, E; k_0, k_1) > 0$. Therefore inference-time computation may produce deliberative gain without producing intelligence under the update definition. □

Definition 9 (Inference-generated update signal)**.** Let $e_E^{(k)}$ denote an update signal produced by system S from behavioural state π_E through k units of inference-time computation. The resulting update is

$$U(\pi_E, e_E^{(k)}) = \pi_{E+\Delta E}.$$

Here ΔE denotes the total energy attributed to the improvement cycle, including the energy required to generate $e_E^{(k)}$ and the energy required to incorporate it through U.

Definition 10 (Interval intelligence at fixed evaluation computation)**.** For an evaluation inference budget k_{eval}, define

$$\mathrm{Int}_S(\mathcal{D}; E, E + \Delta E \mid k_{\text{eval}}) := \frac{\mathbb{E}_{\tau \sim \mathcal{D}} \left[C_{\text{perf}}(\pi_E, k_{\text{eval}}, \tau) - C_{\text{perf}}(\pi_{E+\Delta E}, k_{\text{eval}}, \tau) \right]}{\Delta E}.$$

This is not a rival definition of intelligence. It is the interval definition applied to performance cost while holding the evaluation computation k_{eval} fixed.

Proposition 11.2 (Amortised deliberation constitutes interval intelligence)**.** *Suppose* $e_E^{(k_{\text{gen}})}$ *is produced by inference-time computation from* π_E*, and suppose*

$$U(\pi_E, e_E^{(k_{\text{gen}})}) = \pi_{E+\Delta E}$$

with $\Delta E > 0$*. If, for some fixed evaluation budget* k_{eval}*,*

$$\mathbb{E}_{\tau\sim\mathcal{D}}\left[C_{\text{perf}}(\pi_{E+\Delta E}, k_{\text{eval}}, \tau)\right] < \mathbb{E}_{\tau\sim\mathcal{D}}\left[C_{\text{perf}}(\pi_E, k_{\text{eval}}, \tau)\right],$$

then the reason-update cycle has positive interval intelligence relative to $\mathcal{D}$ *at evaluation budget* k_{eval}*.*

Proof. By definition,

$$\text{Int}_S(\mathcal{D}; E, E+\Delta E \mid k_{\text{eval}}) = \frac{\mathbb{E}_{\tau\sim\mathcal{D}}\left[C_{\text{perf}}(\pi_E, k_{\text{eval}}, \tau) - C_{\text{perf}}(\pi_{E+\Delta E}, k_{\text{eval}}, \tau)\right]}{\Delta E}.$$

The numerator is positive by hypothesis, and $\Delta E > 0$. Therefore the interval intelligence of the cycle is positive. □

Proposition 11.3 (Distillation of deliberative gain)**.** *Fix* $k_{\text{gen}} > k_{\text{eval}}$*. Suppose inference-time computation at budget* k_{gen} *improves expected performance over evaluation budget* k_{eval} *by more than* ε*:*

$$\mathbb{E}_{\tau\sim\mathcal{D}}\left[C_{\text{perf}}(\pi_E, k_{\text{eval}}, \tau) - C_{\text{perf}}(\pi_E, k_{\text{gen}}, \tau)\right] > \varepsilon.$$

Suppose also that updating on the inference-generated signal $e_E^{(k_{\text{gen}})}$ *distils the high-budget behaviour into the new state up to error* ε*:*

$$\mathbb{E}_{\tau\sim\mathcal{D}}\left[C_{\text{perf}}(\pi_{E+\Delta E}, k_{\text{eval}}, \tau) - C_{\text{perf}}(\pi_E, k_{\text{gen}}, \tau)\right] \le \varepsilon.$$

Then

$$\text{Int}_S(\mathcal{D}; E, E+\Delta E \mid k_{\text{eval}}) > 0.$$

Proof. Combining the two inequalities gives

$$\mathbb{E}_{\tau\sim\mathcal{D}}\left[C_{\text{perf}}(\pi_E, k_{\text{eval}}, \tau) - C_{\text{perf}}(\pi_{E+\Delta E}, k_{\text{eval}}, \tau)\right] > 0.$$

Dividing by $\Delta E > 0$ yields

$$\text{Int}_S(\mathcal{D}; E, E+\Delta E \mid k_{\text{eval}}) > 0.$$

□

Proposition 11.4 (Failure under excessive distillation error)**.** *Under the setup above, suppose the distillation error*

$$\mathbb{E}_{\tau\sim\mathcal{D}}\left[C_{\text{perf}}(\pi_{E+\Delta E}, k_{\text{eval}}, \tau) - C_{\text{perf}}(\pi_E, k_{\text{gen}}, \tau)\right]$$

is at least as large as the deliberative gain

$$\mathbb{E}_{\tau\sim\mathcal{D}}\left[C_{\text{perf}}(\pi_E, k_{\text{eval}}, \tau) - C_{\text{perf}}(\pi_E, k_{\text{gen}}, \tau)\right].$$

Then the reason-update cycle has non-positive interval intelligence at k_{eval}*.*

Proof. The stated inequality is equivalent to

$$\mathbb{E}_{\tau\sim\mathcal{D}}\left[C_{\text{perf}}(\pi_E, k_{\text{eval}}, \tau) - C_{\text{perf}}(\pi_{E+\Delta E}, k_{\text{eval}}, \tau)\right] \leq 0.$$

Dividing by $\Delta E > 0$ gives

$$\text{Int}_S(\mathcal{D}; E, E + \Delta E \mid k_{\text{eval}}) \leq 0.$$

□

Remark 4. The separation between k-indexed performance improvement and E-indexed state improvement is especially sharp for frozen AI systems. A frozen model may have $G_S(\mathcal{D}, E; k_0, k_1) > 0$ while $\lambda_S(\tau, E) = 0$, because inference-time computation changes outputs without changing π_E. Human agents typically do not instantiate this separation exactly: deliberation, memory formation, and behavioural update occur together. In such cases the fixed-state frontier is an approximation rather than a literal decomposition.

Example 11.3 (Human agents)**.** Let S be a human agent. Even if S has low current competence on a broad task distribution $\mathcal{D}$, it will usually not be update-static. During practice, study, trial-and-error, or ordinary interaction, the human behavioural state changes through memory formation, skill acquisition, and other biological update processes. Formally, over some interval $[E_0, E_1]$, it is typically possible that

$$\pi_{E_1} \neq \pi_{E_0}$$

and for at least some tasks τ,

$$C(\pi_{E_1}, \tau) < C(\pi_{E_0}, \tau).$$

When this cost decrease holds in expectation over $\mathcal{D}$, the human has positive interval intelligence:

$$\text{Int}_S(\mathcal{D}; E_0, E_1) > 0.$$

This can be true even when current competence is low. A novice, a child, or a generally poor performer may nevertheless have positive intelligence in this framework if energy-consuming interaction reduces future task cost. Conversely, a human with high current competence but no further learning over the interval would have low or zero measured intelligence over that interval.

Example 11.4 (Amortising high-budget reasoning)**.** Suppose a frozen model state π_E has a large deliberative gain between k_{eval} and k_{gen}:

$$G_S(\mathcal{D}, E; k_{\text{eval}}, k_{\text{gen}}) > 0.$$

By itself this is not intelligence. Now suppose the system uses high-budget reasoning at k_{gen} to generate an update signal $e_E^{(k_{\text{gen}})}$, and then applies

$$U(\pi_E, e_E^{(k_{\text{gen}})}) = \pi_{E+\Delta E}.$$

If the updated state performs better at the original evaluation budget,

$$\mathbb{E}_{\tau\sim\mathcal{D}}\left[C_{\text{perf}}(\pi_{E+\Delta E}, k_{\text{eval}}, \tau)\right] < \mathbb{E}_{\tau\sim\mathcal{D}}\left[C_{\text{perf}}(\pi_E, k_{\text{eval}}, \tau)\right],$$

then the deliberation has been amortised into the behavioural state. The composite process has positive interval intelligence at fixed evaluation computation, even though the high-budget reasoning step alone was not intelligence.

Inference-time reasoning is therefore not intelligence when it merely moves a system along a fixed-state performance frontier. However, when the products of such reasoning are used as update signals for U, the composite reason-update cycle instantiates interval intelligence whenever it lowers expected future performance cost at fixed evaluation computation.

Empirical scope. Measurement of amortised-deliberation intelligence is clean in ablation settings where chain-of-thought generation, scoring, and weight update can be isolated and energy-attributed. In production training pipelines, energy attribution to specific cycle components admits only bracketed bounds: baseline energy, generation energy, and update energy are not cleanly separable from one another, and conventional accounting choices propagate into the bounds. The framework therefore predicts that the most informative empirical tests of amortised deliberation will come from controlled distillation experiments where the cycle is instrumented end-to-end, rather than from after-the-fact instrumentation of deployed pipelines. This is the same situation thermodynamic accounting occupies in biology, where bomb-calorimetry of glucose is clean and the energy budget of a forager-day is bounded but not clean; the latter does not impeach the former.

Remark 5 (Exogeneity and energy attribution)**.** The amortised-deliberation analysis depends on the ability to attribute energy to distinct operations on the behavioural state — energy spent generating an inference signal, energy spent incorporating it through U, energy spent on baseline operation. This attribution is licensed because energy is exogenous to the learning process: joules are drawn from outside the system and can be allocated to operations on the state without themselves being part of the state. Experience-indexed frameworks do not naturally support this decomposition, since experience is partly constitutive of the state being updated, and "experience used in inference" versus "experience updated by training" lacks a clean separation. The amortised-deliberation analysis is therefore not merely enabled by the energy denominator; it is one of the places where the exogeneity of the denominator does substantive work.

Remark 6 (Recursive self-improvement)**.** Recursive self-improvement is the iteration of the amortised deliberation cycle: the system distils high-budget inference products into its state via U, and the improved state serves as the starting point for the next cycle. Each iteration has positive interval intelligence at fixed evaluation computation if and only if (i) the deliberative gain $G_S(\mathcal{D}, E_n; k_{\text{eval}}, k_{\text{gen}})$ remains positive, and (ii) the distillation error remains smaller than the gain (Propositions 11.2–11.4). The framework contains no structural barrier to indefinite iteration; whether the conditions hold for a given system is an engineering question about architecture and training, not a consequence of the definitions. The cycle is bounded from below by the irreducible cost floor and from above by available energy. See Appendix A.4 for further discussion.

Remark 7. A final observation on the choice of denominator. Frameworks that index intelligence by experience, data, or sample count measure something real and recover many of the same intuitions — that learning is efficiency, that competence is distinct from the rate at which competence improves, that narrow evaluation rewards memorisation.

The substantive disagreement is structural rather than philosophical. Experience-indexed frameworks require a unit of experience that is specific to the kind of system being evaluated; this is workable within a fixed benchmark but provides no principled cross-substrate comparison, and provides no clean treatment of inference-time computation in modern AI systems, where the question of whether a deliberation step counts as "experience" has no obvious answer.

> Energy-indexed measurement handles both: joules are joules, and the framework's update/performance distinction tells you when an inference-time joule counts toward intelligence (when its product is amortised into the behavioural state through U) and when it does not (when it merely traverses a fixed-state frontier). The amortised-deliberation analysis is the most direct payoff of the energy choice and has, as far as we are aware, no counterpart in experience-indexed frameworks.

12 LLM-Based Agents

Large language models, once their weights are frozen, are artifacts in the sense of Section 6: they sit at a fixed point on some performance frontier and consume energy only for inference, not for self-modification. Under the present framework they possess competence — potentially very high competence — but not intelligence, because no energy they consume during inference updates their behavioural state. Yet systems built *around* frozen LLMs — commonly called agents — often appear intelligent in the everyday sense: they solve novel problems, adapt their strategies across episodes, and improve with use. This section examines why, and identifies the structural feature that separates agents capable of genuine intelligence from those that merely deploy high competence.

The key observation is that an agent is not an LLM. Nor is it merely conventional software wrapping an LLM. It is a composite dynamical system whose state includes not only the LLM's frozen weights but also a mutable *harness* — the tools, memory stores, retrieved context, prompt scaffolding, and workflow logic that mediate between the LLM and the environment. When the harness can be modified by the system's own activity, the composite system acquires a channel for self-modification that the bare LLM lacks. Whether an agent has intelligence, and how much, depends entirely on whether and how efficiently it uses energy to update its harness in ways that reduce future task cost.

This explains both why agents *feel* intelligent and why not all agents are equally so. An agent backed by a capable LLM inherits that model's broad competence, creating an impression of adaptiveness even when the agent's operating procedure is entirely fixed. But two agents built on the same LLM can differ sharply in intelligence if one can restructure its harness and the other cannot.

12.1 Definitions and properties

> **Definition 11** (Agent)**.** An *agent* is a triple $\mathcal{A} = (\pi_E, \mathcal{H}, U_{\mathcal{H}})$, where:
>
> - π_E is a static policy (e.g. a frozen LLM with weights E),
> - $\mathcal{H}$ is a *harness*: the mutable collection of tools, memory, retrieved context, prompt templates, and workflow logic available to π_E, and
> - $U_{\mathcal{H}}$ is the *harness-update operator*: a mapping that, given the current harness and the agent's interaction history, produces a revised harness.
>
> When $U_{\mathcal{H}} = \mathrm{id}$ (the identity), the harness is immutable and the agent reduces to a fixed-policy system.

Definition 12 (Agent effective policy)**.** The *effective policy* of an agent is

$$\pi_{\mathcal{A}} \;=\; \varphi(\pi_E,\, \mathcal{H}),$$

where φ denotes the coupling between the frozen model and the current harness state. Because $\mathcal{H}$ is mutable, $\pi_{\mathcal{A}}$ can change over time even though π_E does not.

Remark 8 (Locus of state change)**.** In a conventional learning system the locus of state change is the model's parameters (weights, synapses, etc.). In an LLM-based agent the model parameters are frozen; the locus of state change is the harness $\mathcal{H}$. Intelligence, where it exists in such a system, is *harness intelligence*: cost reduction achieved by updating $\mathcal{H}$, not π_E.

Proposition 12.1 (Decomposition of agent intelligence)**.** *Let $\mathcal{A} = (\pi_E, \mathcal{H}, U_{\mathcal{H}})$ be an agent with effective policy $\pi_{\mathcal{A}} = \varphi(\pi_E, \mathcal{H})$. The interval intelligence of $\mathcal{A}$ over $[E_0, E_1]$ is*

$$\mathrm{Int}_{\mathcal{A}}(\mathcal{D};\, E_0,\, E_1) \;=\; \frac{\mathbb{E}_{\tau\sim\mathcal{D}}\big[C(\varphi(\pi_{E_0},\, \mathcal{H}_0),\, \tau) - C(\varphi(\pi_{E_1},\, \mathcal{H}_1),\, \tau)\big]}{E_1 - E_0},$$

where π_{E_0}, π_{E_1} are the model states and $\mathcal{H}_0, \mathcal{H}_1$ are the harness states at E_0, E_1 respectively. This is positive whenever the expected cost under the updated effective policy is lower than under the original, regardless of whether the reduction was achieved by updating π_E, updating $\mathcal{H}$, or both. In particular:

(i) If $\pi_{E_0} = \pi_{E_1}$ (frozen model) and $\mathcal{H}_1 \neq \mathcal{H}_0$, then $\mathrm{Int}_{\mathcal{A}} > 0$ if and only if the harness update alone reduces expected cost:

$$\mathbb{E}_{\tau\sim\mathcal{D}}\big[C(\varphi(\pi_E,\, \mathcal{H}_1),\, \tau)\big] \;<\; \mathbb{E}_{\tau\sim\mathcal{D}}\big[C(\varphi(\pi_E,\, \mathcal{H}_0),\, \tau)\big].$$

(ii) If $\mathcal{H}_0 = \mathcal{H}_1$ (frozen harness) and $\pi_{E_1} \neq \pi_{E_0}$, then $\mathrm{Int}_{\mathcal{A}} > 0$ if and only if the model update alone reduces expected cost.

(iii) If both change, the contributions need not decompose additively — the coupling φ may produce interaction effects — but positivity of the total is sufficient for positive intelligence.

Proof. The effective policy at energy E is $\pi_{\mathcal{A}}(E) = \varphi(\pi_E, \mathcal{H}_E)$. Interval intelligence is defined as net expected cost reduction divided by energy consumed (Definition 6). The numerator $\mathbb{E}[C(\pi_{\mathcal{A}}(E_0), \tau) - C(\pi_{\mathcal{A}}(E_1), \tau)]$ depends on the effective policy only through its values at the interval endpoints. Since $\pi_{\mathcal{A}}$ is determined jointly by π_E and $\mathcal{H}$, any change in either component that lowers expected cost makes the numerator positive.

For (i): if π_E is constant, then $\pi_{\mathcal{A}}(E_0) = \varphi(\pi_E, \mathcal{H}_0)$ and $\pi_{\mathcal{A}}(E_1) = \varphi(\pi_E, \mathcal{H}_1)$, so the numerator reduces to $\mathbb{E}[C(\varphi(\pi_E, \mathcal{H}_0), \tau) - C(\varphi(\pi_E, \mathcal{H}_1), \tau)]$, which is positive iff the harness update reduces expected cost. Case (ii) is symmetric. For (iii), note that in general $C(\varphi(\pi_{E_1}, \mathcal{H}_1), \tau) \neq C(\varphi(\pi_{E_1}, \mathcal{H}_0), \tau) + C(\varphi(\pi_{E_0}, \mathcal{H}_1), \tau) - C(\varphi(\pi_{E_0}, \mathcal{H}_0), \tau)$, so contributions from model and harness updates are not additively separable. □

Corollary 2. *An agent can exhibit positive intelligence and increasing competence without any update to its underlying model π_E. Harness updates alone are a sufficient mechanism for intelligence.*

12.2 Examples

Example 12.1 (Fixed-harness agent: customer support)**.** Consider an agent $\mathcal{A}_{\mathrm{CS}}$ deployed for customer support. It has access to a CRM, a knowledge base, and customer notes via tool calls, all orchestrated by a fixed prompt template. For any incoming query τ, the agent routes through the same pipeline: retrieve context $\to$ invoke π_E $\to$ call the appropriate tool $\to$ respond. The harness $\mathcal{H}$ does not change between interactions: the same tools, the same prompt, the same retrieval procedure. Hence $U_{\mathcal{H}} = \mathrm{id}$.
The agent may be highly competent — it resolves tickets quickly and accurately — but the energy it consumes on the nth instance of query τ is the same as on the first. No energy expenditure produces a harness update that reduces future cost. Under the present framework, $\mathrm{Int}_{\mathcal{A}_{\mathrm{CS}}} = 0$.

Example 12.2 (Mutable-harness agent: coding with tool synthesis)**.** Consider an agent $\mathcal{A}_{\mathrm{code}}$ that operates on a codebase and can both write code and register new tools in its own harness. Let $C(\pi_{\mathcal{A}}, \tau)$ denote the cost of performing task τ under the agent's current effective policy — for concreteness, measured in joules of inference energy consumed.
Before any harness update, the agent's effective policy is $\pi_{\mathcal{A}} = \varphi(\pi_E, \mathcal{H})$. On first encountering a recurring refactoring pattern τ, the agent must invoke π_E with full reasoning effort to produce a solution, incurring cost $C(\pi_{\mathcal{A}}, \tau) = C_0$.
The agent then invests energy in a harness update. This proceeds in two stages, which together constitute an instance of amortised deliberation (Section 11.4):

1. **Deliberation.** The agent expends energy E_{k_1} reasoning through τ, producing both a direct solution and an understanding of the task's repeatable structure.

2. **Amortisation.** The agent expends additional energy E_{k_2} to distil that understanding into a reusable code tool t_τ and register it in the harness: $\mathcal{H}' = U_{\mathcal{H}}(\mathcal{H}) = \mathcal{H} \cup \{t_\tau\}$.

The total update energy is $E_U = E_{k_1} + E_{k_2}$, and the agent's effective policy has changed: $\pi'_{\mathcal{A}} = \varphi(\pi_E, \mathcal{H}')$. On all subsequent encounters with τ, the agent invokes t_τ directly, incurring cost $C(\pi'_{\mathcal{A}}, \tau) = C_1 \ll C_0$. The harness has changed but the LLM has not; the cost reduction is carried entirely by the updated harness state.
Substituting into the interval intelligence definition (Definition 6):

$$\mathrm{Int}_{\mathcal{A}}(\{\tau\};\, 0,\, E_U) \;=\; \frac{C_0 - C_1}{E_U} \;=\; \frac{C_0 - C_1}{E_{k_1} + E_{k_2}}.$$

This is positive whenever $C_1 < C_0$, i.e. whenever the synthesised tool actually reduces the cost of τ. The agent's intelligence with respect to τ is higher when (i) the cost drop $C_0 - C_1$ is large — the tool is effective — and (ii) the update energy $E_{k_1} + E_{k_2}$ is small — the tool is cheap to produce. An agent that requires extensive deliberation to build a marginally useful tool has low intelligence; one that quickly synthesises a highly effective tool has high intelligence, even though both operate on the same frozen LLM.

Example 12.3 (Partially mutable harness: memory-augmented agent)**.** An agent $\mathcal{A}_{\mathrm{mem}}$ equipped with a persistent memory store can write observations and summaries that it retrieves in future episodes. This modifies $\mathcal{H}$ (the retrieval context changes), so $U_{\mathcal{H}} \neq \mathrm{id}$. Whether this constitutes intelligence depends on whether the memory updates actually reduce future task cost. If the agent accumulates useful memories that allow it to resolve similar queries with less reasoning effort (fewer tokens, fewer tool calls), it has positive

> intelligence. If the memory grows but lookup cost offsets any saving — or the memories are never usefully retrieved — the net cost reduction may be zero or negative, and the agent's intelligence with respect to $\mathcal{D}$ is correspondingly zero or negative (the latter representing an agent that is making itself worse).

The agent framework is an instance of the amortised-deliberation analysis of Section 11.4. The harness update $U_{\mathcal{H}}$ plays the role of the update operator U: it takes the product of inference-time computation and distils it back into the system's behavioural state — except that the "state" being modified is the harness rather than the model's weights. This makes LLM-based agents the clearest contemporary example of a system in which inference and learning can be distinguished despite the absence of any traditional training step.

12.3 Apparent and genuine intelligence

The distinction between competence and intelligence (Section 9) takes on particular practical importance in the context of LLM-based agents, because the surface behaviour of a high-competence agent is easily mistaken for intelligence.

> **Remark 9** (The appearance of intelligence in fixed-harness agents)**.** A frozen LLM produces varied, context-sensitive, seemingly adaptive outputs. It handles novel queries, adjusts tone and strategy to context, and exhibits what appears to be transfer across domains. An agent built on such a model inherits all of this surface-level adaptiveness. Three features of LLM-based agents are routinely mistaken for intelligence:
>
> (i) *Stochastic variation.* Sampling from the model's output distribution produces different responses to the same query, creating an appearance of exploration or creativity. But variation across samples is a property of the fixed policy π_E, not evidence of state change.
>
> (ii) *In-context learning.* Within a single session, the model conditions on the growing context window, adapting its behaviour to earlier turns. This is deliberative gain (Section 11.4): the effective computation budget k grows with context length, and performance may improve. But unless the session's products are written back into the harness via $U_{\mathcal{H}}$, the gain vanishes when the session ends. The nth session starts at the same cost as the first.
>
> (iii) *Broad training coverage.* The model's pre-training encompassed diverse domains, so it handles unfamiliar queries with surprising adequacy. This is breadth of competence, not generalisation acquired through the agent's own energy expenditure.
>
> None of these constitutes intelligence under the present framework. Each is a property of a fixed effective policy $\pi_{\mathcal{A}} = \varphi(\pi_E, \mathcal{H})$ that does not change between episodes. An agent that gives a brilliant answer to the thousandth instance of a query but consumes the same energy as on the first has $\mathrm{Int}_{\mathcal{A}} = 0$, regardless of how impressive the answers are.

The practical consequence is a test: observe the agent across repeated encounters with the same task distribution. If per-task cost (measured in energy, latency, error, or whatever cost functional is specified) does not fall, the agent has zero intelligence over the observation interval, no matter how capable it appears in any single episode. Competence without trajectory is not intelligence.

12.4 Recursive harness improvement

The coding agent of Example 12.2 builds a single tool and registers it in its harness. A stronger form of the same mechanism arises when the harness update reduces the cost of *future harness updates* — that is, when the agent improves its own capacity to improve.

Definition 13 (Harness-update cost)**.** Let $\mathcal{A} = (\pi_E, \mathcal{H}, U_\mathcal{H})$ be an agent. For a task τ encountered for the first time, define the *harness-update cost* $E_U(\tau, \mathcal{H})$ as the energy the agent must expend to produce a harness revision $\mathcal{H}' = U_\mathcal{H}(\mathcal{H})$ such that $C(\varphi(\pi_E, \mathcal{H}'), \tau) < C(\varphi(\pi_E, \mathcal{H}), \tau)$. This is the price of a single intelligence-producing update.

Example 12.4 (Tool-building tools)**.** Consider $\mathcal{A}_{\text{code}}$ from Example 12.2, extended as follows. After building several task-specific tools, the agent recognises a common pattern and constructs a meta-tool t_{meta} — a code-generation template or scaffold — that automates the boilerplate of tool synthesis. The agent registers t_{meta} in its harness: $\mathcal{H}'' = \mathcal{H}' \cup \{t_{\text{meta}}\}$. Before the meta-tool, building a new task-specific tool for task τ' costs $E_U(\tau', \mathcal{H}') = E'_{k_1} + E'_{k_2}$. After the meta-tool, the same operation costs $E_U(\tau', \mathcal{H}'') = E'_{k_1} + E''_{k_2}$, where $E''_{k_2} \ll E'_{k_2}$ because t_{meta} handles the scaffolding. The meta-tool has reduced the harness-update cost itself.
The consequence for intelligence: for any subsequent task τ', the denominator E_U in the interval intelligence expression shrinks while the numerator $C_0 - C_1$ remains comparable. The agent's intelligence *per task encountered* increases — not because the LLM improved, but because the harness became a better platform for its own extension.

This is recursive self-improvement through the harness channel. We can frame it using the deliberative-gain machinery of Section 11.4:

Proposition 12.2 (Recursive harness improvement as iterated amortised deliberation)**.** *Let $\mathcal{A} = (\pi_E, \mathcal{H}_n, U_\mathcal{H})$ denote the agent at the nth iteration, with π_E fixed throughout. At each iteration:*

1. *The agent performs high-budget deliberation at cost k_{gen}, generating a candidate harness modification δ_n.*
2. *The modification is applied: $\mathcal{H}_{n+1} = U_\mathcal{H}(\mathcal{H}_n, \delta_n)$.*
3. *The resulting effective policy $\varphi(\pi_E, \mathcal{H}_{n+1})$ has lower expected cost than $\varphi(\pi_E, \mathcal{H}_n)$ on $\mathcal{D}$.*

Each iteration has positive interval intelligence if and only if the deliberative gain G at stage (1) is positive and the amortisation at stage (2) preserves enough of that gain (cf. Propositions 11.2–11.4).
The recursion is self-improving *when the harness modification at round n reduces the update cost at round $n+1$: $E_U(\tau, \mathcal{H}_{n+1}) < E_U(\tau, \mathcal{H}_n)$ for tasks $\tau \sim \mathcal{D}$. In this regime, the intelligence of successive iterations increases even at constant deliberation quality, because the denominator of the intelligence ratio shrinks. The process is bounded from below by the irreducible cost floor and from above by available energy, as in the general RSI analysis (Remark 6).*

Remark 10 (Near-term recursive self-improvement)**.** Recursive harness improvement is arguably the most likely near-term form of recursive self-improvement. Traditional RSI requires a system to modify its own model weights — a capability that few deployed systems possess and that raises significant safety concerns. Harness-channel RSI requires only that the agent can modify its own tools, memory, or workflow — capabilities that many current agent frameworks already provide. The framework's decomposition (Proposition 12.1) makes this precise: a frozen π_E with a recursively improving $\mathcal{H}$ is already undergoing RSI, even though no weights have changed.

13 Parameters Left to Convention

The framework deliberately exposes four parameters rather than fixing them.

1. **The boundary between S and its producing process ρ.** Energy is a clean physical quantity given a system boundary; the boundary itself is a convention. Reasonable conventions exist for most cases: humans begin at birth or some comparable biological landmark, AI systems begin at the start of their own training. Boundary cases, such as continuous fine-tuning, distillation, and model merging, require explicit choices.
2. **The breadth of the task distribution $\mathcal{D}$.** Universe-$\mathcal{D}$ is the theoretical ideal. Practical evaluations restrict $\mathcal{D}$ and inherit weighting and gaming risks proportional to that restriction. Below a certain breadth the metric stops measuring intelligence and starts measuring memorisation or domain competence.
3. **The cost weights in C.** The cost functional schema does not prescribe how time, effort, risk, and error trade off. This is left to the evaluator and is domain-dependent.
4. **The evaluation computation budget k_{eval} (where amortised deliberation is evaluated).** Section 11.4's interval intelligence at fixed evaluation computation is well-defined only once k_{eval} is specified. Different evaluation budgets give different rankings for the same system and update interval; this is intended, since different budgets ask different questions about the post-update state.

Together, these parameters constitute the *specification* relative to which intelligence values are defined (see Section 2). The framework's claim is to well-defined substrate-neutral comparison within a specification, not to specification-free intelligence values. Cross-specification comparisons require explicit translation between specifications and are not automatically licensed.

14 Parameters Left to Convention

The framework deliberately exposes four parameters rather than fixing them.

1. **The boundary between S and its producing process ρ.** Energy is a clean physical quantity given a system boundary; the boundary itself is a convention. Reasonable conventions exist for most cases: humans begin at birth or some comparable biological landmark, AI systems begin at the start of their own training. Boundary cases, such as continuous fine-tuning, distillation, and model merging, require explicit choices.
2. **The breadth of the task distribution $\mathcal{D}$.** Universe-$\mathcal{D}$ is the theoretical ideal. Practical evaluations restrict $\mathcal{D}$ and inherit weighting and gaming risks proportional to that restriction. Below a certain breadth the metric stops measuring intelligence and starts measuring memorisation or domain competence.

3. **The cost weights in C.** The cost functional schema does not prescribe how time, effort, risk, and error trade off. This is left to the evaluator and is domain-dependent.
4. **The evaluation computation budget k_{eval} (where amortised deliberation is evaluated).** Section 11.4's interval intelligence at fixed evaluation computation is well-defined only once k_{eval} is specified. Different evaluation budgets give different rankings for the same system and update interval; this is intended, since different budgets ask different questions about the post-update state.

Together, these parameters constitute the *specification* relative to which intelligence values are defined (see Section 2). The framework's claim is to well-defined substrate-neutral comparison within a specification, not to specification-free intelligence values. Cross-specification comparisons require explicit translation between specifications and are not automatically licensed.

A Properties of the Definitions

This appendix records formal properties of the preceding definitions. Throughout, $\mathcal{T}$ is a task space, $\mathcal{D}$ is a probability distribution over $\mathcal{T}$, and S is a system with behavioural state π_E after cumulative update-energy $E \geq 0$. The expected cost of S on task τ after update-energy E is written

$$C_S(E, \tau) := C(\pi_E, \tau) \geq 0.$$

The local learning rate on τ is

$$\lambda_S(\tau, E) := -\frac{\mathrm{d}C_S(E, \tau)}{\mathrm{d}E},$$

and local intelligence relative to $\mathcal{D}$ is

$$\mathrm{Int}_S(\mathcal{D}, E) := \mathbb{E}_{\tau\sim\mathcal{D}}\left[-\frac{\mathrm{d}C_S(E, \tau)}{\mathrm{d}E}\right].$$

For empirical work it is often preferable to use an interval form. For $E_0 < E_1$, define

$$\mathrm{Int}_S(\mathcal{D}; E_0, E_1) := \frac{\mathbb{E}_{\tau\sim\mathcal{D}}\left[C_S(E_0, \tau) - C_S(E_1, \tau)\right]}{E_1 - E_0}.$$

This interval expression is the measurable finite-budget analogue of the differential definition.

A.1 Properties of Competence

Proposition A.1 (Competence rankings are invariant under monotone reparameterisation — proof)**.** *Since g is strictly decreasing, $x < y$ if and only if $g(x) > g(y)$. Applying this with $x = C(\pi_E^1, \tau)$ and $y = C(\pi_E^2, \tau)$ yields the result stated in Proposition 9.1.*

Proposition A.2 (High competence does not imply intelligence)**.** *There exists a system S with arbitrarily high competence on $\mathcal{D}$ but zero intelligence on $\mathcal{D}$.*

Proof. Let S be update-static over all $E \geq 0$, meaning $\pi_E = \pi_0$ for all $E \geq 0$. Then $C_S(E, \tau) = C(\pi_0, \tau)$ is constant in E, so $\lambda_S(\tau, E) = 0$ for every τ, and hence $\mathrm{Int}_S(\mathcal{D}, E) = 0$. However, if S is a static lookup table with $C_S(E, \tau) \leq \varepsilon$ for every τ in the support of $\mathcal{D}$, competence can be made arbitrarily high by choosing ε arbitrarily small. Intelligence remains zero because the behavioural state never updates. □

Proposition A.3 (Intelligence does not imply high current competence)**.** *There exists a system S with positive intelligence on $\mathcal{D}$ but low current competence.*

Proof. Let $C_S(E, \tau) = M - aE$ for $E \in [0, M/a]$, with $M > 0$ and $a > 0$. Then $\mathrm{Int}_S(\mathcal{D}, E) = a > 0$. At $E = 0$, $C_S(0, \tau) = M$. For arbitrarily large M, current competence is arbitrarily low, yet intelligence is positive. □

Competence is a level; intelligence is a slope.

Remark 11 (Aggregation conceals competence profile differences)**.** Two systems may have the same expected competence over $\mathcal{D}$ while having different competence profiles: one uniformly moderate across tasks, another excellent on some tasks and poor on others. The scalar quantity $-\mathbb{E}_{\tau \sim \mathcal{D}}[C(\pi_E, \tau)]$ records average competence, not robustness, worst-case competence, or variance across tasks. Where profile shape matters, empirical evaluations should report the distribution of costs across $\mathcal{D}$, not only the expectation.

A.2 Properties of Learning

Proposition A.4 (Learning on one task need not imply learning on another — proof)**.** *Let π_{E_0} assign equal probability to two actions a_1, a_2, where a_1 is optimal for τ_1 and a_2 is optimal for τ_2. Let the update specialise the policy toward a_1, increasing its probability. Then expected cost on τ_1 falls and expected cost on τ_2 rises. This establishes Proposition 9.2.*

Proposition A.5 (The sign of learning is ordinally invariant; the magnitude is not — proof)**.** *Strict monotonicity of f preserves strict order: $x < y$ if and only if $f(x) < f(y)$. For magnitudes, $f(C(E_0)) - f(C(E_1))$ need not equal $a[C(E_0) - C(E_1)]$ for any constant a unless f is affine. This establishes Proposition 9.3.*

Proposition A.6 (Interval learning is additive across adjacent intervals)**.** *For $E_0 < E_1 < E_2$,*

$$L_S(\tau; E_0, E_2) = L_S(\tau; E_0, E_1) + L_S(\tau; E_1, E_2).$$

Hence net learning over a long interval decomposes exactly into net learning over subintervals.

Proof.

$$C_S(E_0, \tau) - C_S(E_2, \tau) = [C_S(E_0, \tau) - C_S(E_1, \tau)] + [C_S(E_1, \tau) - C_S(E_2, \tau)].$$

□

Proposition A.7 (Efficient memorisation can fail to transfer)**.** *There exists a system S and task distributions $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ such that S has high apparent intelligence on $\mathcal{D}_{\text{train}}$ but zero transfer intelligence on $\mathcal{D}_{\text{test}}$.*

Proof. Let the task space be finite and partitioned as $\mathcal{T} = \mathcal{T}_{\text{train}} \cup \mathcal{T}_{\text{test}}$ with $\mathcal{T}_{\text{train}} \cap \mathcal{T}_{\text{test}} = \varnothing$. Let $\mathcal{D}_{\text{train}}$ have support only on $\mathcal{T}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ only on $\mathcal{T}_{\text{test}}$. Consider a lookup-style learner that stores exact responses for training tasks, reducing cost from 1 to 0 after energy $\varepsilon > 0$. Then

$\mathrm{Int}_S(\mathcal{D}_{\mathrm{train}}; 0, \varepsilon) = 1/\varepsilon$, which is large for small ε. But if the memorised table has no effect on test tasks, $\mathrm{Int}_S(\mathcal{D}_{\mathrm{test}}; 0, \varepsilon) = 0$. □

This formalises the anti-memorisation pressure behind the breadth requirement on $\mathcal{D}$.

A.3 Properties of Intelligence

Proposition A.8 (Positive affine cost rescaling preserves intelligence rankings)**.** *Let two systems S_1 and S_2 be evaluated under cost C. Define transformed cost $C' = aC + b$, where $a > 0$. Then*

$$\mathrm{Int}_{S_1}(\mathcal{D}, E) > \mathrm{Int}_{S_2}(\mathcal{D}, E) \quad \textit{if and only if} \quad \mathrm{Int}'_{S_1}(\mathcal{D}, E) > \mathrm{Int}'_{S_2}(\mathcal{D}, E).$$

Proof. For any system S, $C'_S(E, \tau) = aC_S(E, \tau) + b$. Differentiating: $-\mathrm{d}C'_S/\mathrm{d}E = a(-\mathrm{d}C_S/\mathrm{d}E)$. Taking expectation over $\mathcal{D}$: $\mathrm{Int}'_S(\mathcal{D}, E) = a\,\mathrm{Int}_S(\mathcal{D}, E)$. Since $a > 0$, multiplication by a preserves strict order. □

Changing units, such as seconds to milliseconds, scales the numerical intelligence value but does not change rankings if the same affine transformation is applied to all systems.

Proposition A.9 (Nonlinear monotone cost transformations need not preserve rankings)**.** *There exist systems S_1, S_2, and a strictly increasing nonlinear transformation f such that S_1 ranks above S_2 under C but below S_2 under $f(C)$.*

Proof. Use a one-task distribution. Let $C_1(E) = 1$ with $-\mathrm{d}C_1/\mathrm{d}E = 2$, and $C_2(E) = 10$ with $-\mathrm{d}C_2/\mathrm{d}E = 1$. Under original cost, $\mathrm{Int}_{S_1} = 2 > 1 = \mathrm{Int}_{S_2}$. Transform by $f(C) = C^2$. The transformed rate is $-\mathrm{d}(C^2)/\mathrm{d}E = 2C(-\mathrm{d}C/\mathrm{d}E)$. For S_1: $2(1)(2) = 4$. For S_2: $2(10)(1) = 20$. Rankings reverse. □

This result is not a defect requiring repair but a structural feature of cost-relativity (see the cost-relativity discussion in Section 8). Nonlinear monotone transformations encode substantively different cost specifications — different attitudes toward how large errors trade off against small ones — rather than different units of the same specification.

Proposition A.10 (Interval intelligence is an energy-weighted average of subinterval intelligence)**.** *For $E_0 < E_1 < E_2$,*

$$\mathrm{Int}_S(\mathcal{D}; E_0, E_2) = \frac{(E_1 - E_0)\,\mathrm{Int}_S(\mathcal{D}; E_0, E_1) + (E_2 - E_1)\,\mathrm{Int}_S(\mathcal{D}; E_1, E_2)}{E_2 - E_0}.$$

Proof. By definition, $(E_2 - E_0)\,\mathrm{Int}_S(\mathcal{D}; E_0, E_2) = \mathbb{E}_{\tau\sim\mathcal{D}}[C_S(E_0, \tau) - C_S(E_2, \tau)]$. By the additivity of net learning (Proposition A.6),

$$\begin{aligned}\mathbb{E}_{\tau\sim\mathcal{D}}[C_S(E_0, \tau) - C_S(E_2, \tau)] &= \mathbb{E}_{\tau\sim\mathcal{D}}[C_S(E_0, \tau) - C_S(E_1, \tau)] + \mathbb{E}_{\tau\sim\mathcal{D}}[C_S(E_1, \tau) - C_S(E_2, \tau)] \\ &= (E_1 - E_0)\,\mathrm{Int}_S(\mathcal{D}; E_0, E_1) + (E_2 - E_1)\,\mathrm{Int}_S(\mathcal{D}; E_1, E_2).\end{aligned}$$

Dividing both sides by $(E_2 - E_0)$ gives the result. □

Proposition A.11 (Interval intelligence converges to local intelligence)**.** *If $C_S(E,\tau)$ is differentiable in E, then*

$$\lim_{\Delta E \to 0^+} \mathrm{Int}_S(\mathcal{D}; E, E+\Delta E) = \mathrm{Int}_S(\mathcal{D}, E),$$

assuming the regularity needed to exchange limit and expectation.

Proof. By definition, $\mathrm{Int}_S(\mathcal{D}; E, E+\Delta E) = \mathbb{E}_{\tau\sim\mathcal{D}}\big[(C_S(E,\tau) - C_S(E+\Delta E,\tau))/\Delta E\big]$. By differentiability, the difference quotient converges to $-\mathrm{d}C_S(E,\tau)/\mathrm{d}E$ for each τ. Under dominated convergence, the limit and expectation commute. □

This justifies finite-difference estimates in empirical applications.

Proposition A.12 (Inference-only computation does not constitute intelligence under the update definition)**.** *Suppose a system spends additional inference energy to improve performance on a task but does not update its behavioural state π. Then this inference computation may improve task performance, but it does not constitute intelligence as cost reduction through update-energy.*

Proof. If inference computation does not modify π_E, then $C(\pi'_E,\tau) = C(\pi_E,\tau)$. The inference procedure varies k (deployment computation), not the E-indexed behavioural state. It therefore does not contribute to $-\mathrm{d}C(\pi_E,\tau)/\mathrm{d}E$. □

Proposition A.13 (Target-threshold intelligence is ill-behaved for already-competent systems)**.** *If target-threshold intelligence is defined as $\mathrm{Int}_S(\tau, C^*) = 1/E_S(\tau, C^*)$, where $E_S(\tau, C^*)$ is the energy required to reach cost C^* or lower, then any system already satisfying the threshold at $E = 0$ has infinite or undefined target-threshold intelligence.*

Proof. If $C_S(0,\tau) \le C^*$, then $E_S(\tau, C^*) = 0$, and $\mathrm{Int}_S(\tau, C^*) = 1/0$. □

Repair. Define target-threshold intelligence only conditionally, when $C_S(0,\tau) > C^*$. Alternatively, define improvement-threshold energy:

$$E_S(\tau, \Delta C) = \inf\{E \ge 0 : C_S(0,\tau) - C_S(E,\tau) \ge \Delta C\}.$$

Then define

$$\mathrm{Int}_S(\tau, \Delta C) = \frac{\Delta C}{E_S(\tau, \Delta C)}.$$

This measures the energy required to produce a given improvement, not merely the energy required to possess a competence level. Relative to a task distribution $\mathcal{D}$,

$$\mathrm{Int}_S(\mathcal{D}, \Delta C) := \mathbb{E}_{\tau\sim\mathcal{D}}\left[\frac{\Delta C}{E_S(\tau, \Delta C)}\right].$$

Proposition A.14 (Wasted energy lowers measured intelligence)**.** *If two systems produce the same expected cost reduction but one consumes more total energy, then the one consuming more energy has lower interval intelligence.*

Proof. Let $\Delta C > 0$ be the shared expected cost reduction. If S_1 consumes energy A and S_2 consumes $B > A > 0$, then $\mathrm{Int}_{S_1} = \Delta C/A > \Delta C/B = \mathrm{Int}_{S_2}$. ☐

Proposition A.15 (If no reducible cost remains, intelligence collapses to zero)**.** *If a system has reached irreducible minimum cost on all tasks in the support of $\mathcal{D}$, then its intelligence is zero even if its competence is maximal.*

Proof. If $C_S(E, \tau) = C^\infty(\tau)$ for all τ in the support of $\mathcal{D}$, then $C_S(E, \tau) - C_S(E + \Delta E, \tau) = 0$ for all $\Delta E > 0$, so $\mathrm{Int}_S(\mathcal{D}; E, E + \Delta E) = 0$. ☐

This is the recovery-from-ignorance character of the definition.

Proposition A.16 (Raw intelligence can reward initial incompetence)**.** *The raw derivative measure can rank a worse system above a better system merely because the worse system has more remaining reducible cost.*

Proof. Let two systems have exponential learning curves with the same coefficient $k > 0$: $C_A(E) = C_A^\infty + A_A \exp(-kE)$ and $C_B(E) = C_B^\infty + A_B \exp(-kE)$, where $A_A > A_B > 0$. Then $-\mathrm{d}C_A/\mathrm{d}E = kA_A \exp(-kE) > kA_B \exp(-kE) = -\mathrm{d}C_B/\mathrm{d}E$. The raw derivative ranks A above B because A has more remaining reducible cost, not because it has a better learning coefficient. ☐

Repair. Use proportional intelligence where an asymptotic cost floor is meaningful:

$$\widetilde{\mathrm{Int}}_S(\mathcal{D}, E) = \mathbb{E}_{\tau\sim\mathcal{D}}\left[-\frac{1}{C_S(E,\tau) - C_S^\infty(\tau)} \frac{\mathrm{d}C_S(E,\tau)}{\mathrm{d}E}\right].$$

For $C_S(E) = C_S^\infty + A\exp(-kE)$, this reduces to k. The proportional measure recovers the learning coefficient rather than rewarding the system for having more remaining reducible cost.

Proposition A.17 (Power-constrained and energy-constrained intelligence coincide under fixed duration)**.** *If systems are evaluated over the same time interval $[t_0, t_1]$, and each has constant power consumption P, then comparing cost reduction per unit energy is equivalent to comparing cost reduction per unit power.*

Proof. If power is constant, $E = P\Delta t$. Then $\mathrm{Int}_S = \Delta C/E = \Delta C/(P\Delta t)$. If Δt is fixed across systems, rankings by $\Delta C/E$ are equivalent to rankings by $\Delta C/P$. ☐

A.4 Recursive Self-Improvement

The amortised deliberation cycle defined in Section 11.4 admits indefinite iteration. At each step n, the system generates high-budget inference products from state π_{E_n}, distils them into $\pi_{E_{n+1}}$ via the update operator U, and the improved state serves as the starting point for step $n + 1$. Each step is a fresh instance of the amortised deliberation analysis, and has positive interval intelligence if and only if two empirical conditions hold:

1. *Sustained deliberative gain*: $G_S(\mathcal{D}, E_n; k_{\mathrm{eval}}, k_{\mathrm{gen}}) > 0$. The system at evaluation budget k_{eval} still performs worse than at generation budget k_{gen}, so there remains a gap to distil.

2. *Faithful distillation*: the distillation error remains smaller than the deliberative gain, so the updated state at k_{eval} captures enough of the high-budget improvement to yield net cost reduction (Propositions 11.2 and 11.4).

No theorem within the framework requires either condition to fail after finitely many cycles. Whether the conditions hold for a given system is determined by its architecture, cost landscape, and the relationship between k_{gen} and k_{eval} — not by the structure of intelligence as defined here. Recursive self-improvement is therefore an engineering and parameter problem rather than a fundamental one.

Two boundaries limit the cycle. The irreducible cost floor bounds it from below: as the system approaches minimum achievable cost $C^{\infty}(\tau)$ on tasks in $\mathcal{D}$, available cost reduction shrinks and the numerator of interval intelligence goes to zero regardless of distillation quality. Energy bounds it from above: each iteration consumes $\Delta E > 0$, so the total number of iterations is bounded by available energy.

Within these bounds, the rate of improvement at each step is an empirical property of the system. The deliberative gain $G_S(\mathcal{D}, E_n; k_{\text{eval}}, k_{\text{gen}})$ is itself a function of the current state π_{E_n}. It could increase with competence — if better states extract more from additional inference, yielding accelerating improvement. It could decrease — if easy gains are harvested first, yielding decelerating improvement. Or it could remain approximately constant over some range. The framework does not prescribe which regime obtains; it identifies the quantities whose trajectory determines which regime is realised.

B Recommended Formal Definitions

The preceding results suggest that the framework should treat the following forms as primary.

1. **Empirical interval intelligence**

$$\mathrm{Int}_S(\mathcal{D}; E_0, E_1) = \frac{\mathbb{E}_{\tau\sim\mathcal{D}}\left[C_S(E_0,\tau) - C_S(E_1,\tau)\right]}{E_1 - E_0}.$$

2. **Transfer intelligence**

$$\mathrm{Int}_S^{\text{transfer}}(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}; E_0, E_1) = \frac{\mathbb{E}_{\tau\sim\mathcal{D}_{\text{test}}}\left[C_S(E_0,\tau) - C_S(E_1,\tau)\right]}{E_1 - E_0},$$

where updates are driven by $\mathcal{D}_{\text{train}}$ but evaluation is performed on $\mathcal{D}_{\text{test}}$. This directly tests whether update-energy purchases generalisation rather than memorisation.

3. **Proportional learning efficiency**

$$\widetilde{\mathrm{Int}}_S(\mathcal{D}, E) = \mathbb{E}_{\tau\sim\mathcal{D}}\left[-\frac{1}{C_S(E,\tau) - C_S^{\infty}(\tau)}\frac{\mathrm{d}C_S(E,\tau)}{\mathrm{d}E}\right],$$

where $C_S^{\infty}(\tau)$ is an interpretable asymptotic or irreducible cost floor.

4. **Improvement-threshold intelligence**

$$E_S(\tau, \Delta C) = \inf\{E \geq 0 : C_S(0,\tau) - C_S(E,\tau) \geq \Delta C\},$$

$$\mathrm{Int}_S(\tau, \Delta C) = \frac{\Delta C}{E_S(\tau, \Delta C)}.$$

This avoids the division-by-zero pathology of competence-threshold definitions for systems that already begin below the target cost.

5. **Amortised deliberation at fixed evaluation computation**

$$\mathrm{Int}_S(\mathcal{D}; E, E+\Delta E \mid k_{\mathrm{eval}}) = \frac{\mathbb{E}_{\tau\sim\mathcal{D}}\left[C_{\mathrm{perf}}(\pi_E, k_{\mathrm{eval}}, \tau) - C_{\mathrm{perf}}(\pi_{E+\Delta E}, k_{\mathrm{eval}}, \tau)\right]}{\Delta E}.$$

This is the interval form applied to a reason-update cycle, with the evaluation computation held fixed so that measured improvement is attributed to update of π_E, not to extra deployment computation.

C Illustrative Cross-Substrate Comparison

This appendix applies the framework to five systems spanning biological and computational substrates. The comparison is illustrative, not empirical in the strict sense: the energy denominators have physical anchors, but the numerators are structured judgements about fractional cost reduction over a universal task distribution. The interest of the exercise is not the specific numerical values but the observation that the energy denominator compresses cross-substrate comparisons when the task distribution is broad — a behaviour the framework predicts but does not guarantee.

C.1 Setup

The simplified estimator is the interval intelligence form:

$$\mathrm{Int}_S = \frac{\Delta C_S}{E_S},$$

where ΔC_S is the estimated fraction of universal task-cost reduction achieved by the system over its update interval, and E_S is total energy consumed over that interval. The universal cost-reduction ceiling is normalised to $\Delta C_{\mathrm{max}} = 1$. Each system receives a fractional numerator representing its estimated contribution.

The five systems are: a common raven (lifetime), a single human (lifetime), a collective of 100 collaborating experts (summed lifetimes), a frontier AI full training run, and a frontier AI post-training-only process. System boundaries follow the conventions of Section 5: energy is attributed to the update process, not to the producing process that preceded it. For biological systems, this means total metabolic energy from birth (or effective juvenile survival) to death. For AI systems, this means wall-plug energy for the training or post-training run, excluding researcher labour, prior model lineage, hardware manufacture, and data production.

C.2 Energy Denominators

System	Energy (J)	Derivation
Common raven	6.6×10^9	$\sim$21 W × 10 yr field metabolic rate
Single human	2.5×10^{11}	$\sim$109 W × 73 yr total metabolic energy
100 experts	2.8×10^{13}	$100 \times 2.8 \times 10^{11}$ J (developed-country lifetime)
AI full training	1.8×10^{14}	$\sim$50 GWh frontier-class training estimate
AI post-training	9.0×10^{13}	$\sim$25 GWh reasoning-focused post-training

The human denominator uses total daily energy expenditure ($\sim$109 W), not brain-only energy ($\sim$20 W), following the framework's convention that all energy consumed by the system counts against intelligence. The 100-expert denominator is a simple sum of individual lifetimes and does not include institutional overhead (buildings, labs, transport, computing infrastructure), which makes it favourable to the collective. The AI denominators reference public anchors for frontier-class training and exclude broader social and industrial inputs.

C.3 Numerator Estimates: Universe-Balanced Distribution

System	ΔC	Rationale
Common raven	3×10^{-9}	Ecological cognition; $\sim 10^{-3}$ of human
Single human	3×10^{-6}	Calibration anchor; language, motor, social, professional
100 experts	1×10^{-3}	$\sim 3.3\times$ super-additive over 100 individuals
AI full training	1×10^{-3}	Broad symbolic competence; penalised for no embodiment
AI post-training	2×10^{-4}	~20% of full-training breadth; alignment, reasoning, formatting

The single human is the calibration anchor: a developed adult acquires language, motor control, social cognition, cultural behaviour, tool use, numeracy, and typically a profession — enormous relative to a newborn, but a tiny fraction of all physically possible tasks. The raven receives $\sim 10^{-3}$ of the human numerator, reflecting substantial ecological cognition without the symbolic-cultural stack. The 100-expert collective assumes a $3.3\times$ super-additivity factor over linear aggregation, reflecting specialisation and coordination gains. The AI full-training numerator matches the expert collective in breadth but is penalised for lacking autonomous embodiment. Post-training receives 20% of the full-training numerator.

These numerators are structured judgements, not measurements. They are the weakest link in the comparison.

C.4 Results and Perturbation Analysis

Under the universe-balanced distribution, the intelligence values and their spread are:

Rank	System	ΔC	Energy (J)	Int_S (J^{-1})
1	100 experts	1.0×10^{-3}	2.8×10^{13}	3.57×10^{-17}
2	Single human	3.0×10^{-6}	2.5×10^{11}	1.20×10^{-17}
3	AI full training	1.0×10^{-3}	1.8×10^{14}	5.56×10^{-18}
4	AI post-training	2.0×10^{-4}	9.0×10^{13}	2.22×10^{-18}
5	Common raven	3.0×10^{-9}	6.6×10^{9}	4.55×10^{-19}
			Spread (max/min):	**79×**

The denominators span $\sim 2.7 \times 10^4$ and the numerators span $\sim 3.3 \times 10^5$, yet the final intelligence values span only $\sim 79\times$. This compression is the central observation: the energy denominator regularises cross-substrate comparison.

To test robustness, the numerators were adjusted under four alternative task distributions while holding denominators fixed:

Distribution	All-system spread	Character
Universe-balanced	79×	Broad mixed; original case
Cognitive-symbolic	353× (9.6× excl. raven)	Language, math, coding
Social-cultural	35×	Communication, norms, teaching
Embodied-physical	900×	Locomotion, manipulation, survival
Scientific-engineering	7,040× (19× excl. raven)	Modelling, experiment, design

Broad distributions that give every system non-trivial support produce compressed spreads (35×–79×). Narrow distributions that reduce one system's numerator to near-zero produce large spreads (353×–7,040×), but the expansion is driven almost entirely by the near-zero system; among the systems meaningfully engaged by the distribution, the spread remains compressed (9.6×–19×).

This is exactly the behaviour the framework predicts. A task distribution too narrow for a system to have meaningful support stops measuring intelligence and starts measuring domain-specific efficiency. The breadth requirement on $\mathcal{D}$ (Section 10) is what prevents the framework from rewarding substrate advantage or penalising domain irrelevance.

C.5 Interpretation and Limitations

The compressed range under broad distributions is not proof that the framework is correct. It is evidence that the definition is not obviously pathological: when plausible inputs are supplied, the energy denominator imposes a regularising pressure that produces non-trivial cross-substrate comparisons rather than trivially substrate-dominated rankings.

Several limitations constrain the comparison. The numerators are hand-built judgements; different calibrators would produce different values. The system boundaries are narrower than a fully ecological accounting would use — wider boundaries (including institutional energy for experts, or hardware manufacture for AI) would change the rankings. The 100-expert super-additivity factor is especially uncertain. The AI post-training landscape is moving rapidly and the assigned values may not reflect current practice.

The strongest conclusion is therefore conditional: *if* the energy denominator is the right normaliser and *if* the numerator judgements are in the right order of magnitude, then cross-substrate intelligence values compress to within roughly two orders of magnitude under broad task distributions. Whether the antecedents hold is an empirical question the framework poses but does not answer.