Thermodynamic, Information-Theoretic, and Epistemic Constraints on Artificial Intelligence Systems

Madhav Moganti
Bell Labs, Nokia · mmoganti@outlook.com
and
Ramana Nyapathi
Aganitha Cognitive Services · ramana@aganitha.ai

Abstract

For a fixed hardware generation and a fixed task-success threshold, broadening the admissible task and input scope is hypothesized to increase energy per verified successful output and the cost of achieving specified verification coverage. Under recursive replacement of reference data by unvalidated generated samples, distributional tail coverage is hypothesized to decline with recursion depth. These effects are predicted to weaken when task-relevant complexity is held constant, original data are retained, generated samples are validated and reweighted, or the system is decomposed into independently testable bounded modules.

Contemporary scaling laws relate cross-entropy loss to model size, data volume, and compute, but do not characterize energy per verified successful task, task-relevant representation requirements, distributional degradation under recursive synthetic-data workflows, or verification coverage. We synthesize three well-documented mechanisms—Landauer's lower bound on irreversible information processing (distinct from practical hardware costs), information-bottleneck notions of minimum representation complexity for a target error, and conditional model-collapse under pure recursive replacement—and place them inside a multi-objective cost-and-risk vector whose components retain distinct physical and statistical units. Domain-bounded hierarchical architectures with typed action interfaces, preserved real-data anchors, and explicit failure containment reduce, but do not eliminate, these risks.

We distinguish universal undecidability results from bounded neural-network verification and examine the practical scalability of formal methods. Abstract-interpretation domains (intervals, zonotopes, DeepPoly/CROWN-style polyhedral bounds, star sets) supply the geometric language of incomplete verifiers; branch-and-bound and cutting-plane methods extend reach; hierarchical and assume-guarantee composition convert monolithic obligations into chains of smaller, contract-linked sub-problems. We supply falsifiable experimental predictions together with enterprise controls based on data provenance, constrained interfaces, and post-deployment monitoring.

Keywords: *thermodynamics of computation; information bottleneck; task-relevant representation; model collapse; synthetic data; formal verification; abstract interpretation; assume-guarantee composition; hierarchical verification; bounded AI; reliable AI systems*

1. Introduction

1.1 Motivation: Scaling Laws and Omitted Observables

Empirical scaling-law studies report approximately power-law relationships between cross-entropy loss and model size, dataset size, and training compute over substantial ranges [10, 11]. These regularities are important, yet they do not imply that every operational quantity improves at the same rate or that scale is cost-free. Benchmark loss is only one observable. Energy per verified successful task, robustness under distribution shift, recursive-data degradation, verification coverage, and coordination cost may follow different scaling relationships. A system can therefore improve on benchmark loss while becoming more expensive to operate, harder to verify, or more sensitive to data lineage.

A verified successful output is an output that satisfies a predeclared task-success criterion together with any applicable calibration, safety, authorization, and verification checks. The exact criteria are application-specific and must be fixed before architectures or scales are compared.

1.2 Central Thesis and Scope

The paper develops three cost mechanisms and one verification boundary: (i) energy dissipation in implemented computation, measured operationally as joules per verified successful task; (ii) task-relevant representation complexity required to meet a specified predictive error; and (iii) distributional drift under recursive synthetic-data replacement. The verification boundary distinguishes universal undecidability from bounded formal verification and open-world specification gaps, and examines the practical scalability of abstract-interpretation domains, branch-and-bound search, hierarchical decomposition, and assume-guarantee composition.

The central claim is not that general-purpose AI must collapse. It is that broadening task and input scope can increase the resources required to achieve a fixed reliability target, and that recursive data exposure, modular objective conflict, and verification difficulty can add independent or interacting risks. These claims are stated as hypotheses with explicit controls and potential falsifiers in Section 9.

1.3 Biological Analogy: What Transfers and What Does Not

Artificial neural networks are not mechanistically equivalent to biological nervous systems. What does transfer is the substrate-independent statement that every physical implementation of computation is subject to thermodynamic constraints. Practical costs arise from logically irreversible state transformations, switching, communication, memory refresh, noise suppression, error correction, and heat removal [1, 2]. The adult human brain's approximate 20-watt power budget [9] is used only as an empirical illustration of capable information processing under a stringent energy constraint; it does not establish proximity to the Landauer limit or a silicon scaling rate.

1.4 Contributions

1. A taxonomy that separates thermodynamic limits, task-relevant representation requirements, recursive-data drift, modular objective conflict, and verification limits.
2. A sharpened domain-scope hypothesis that relates the number and heterogeneity of task-relevant decision distinctions that must be preserved at fixed error to the cost of meeting that error.
3. An architectural account of domain-bounded and hierarchical systems that reduces verification and monitoring scope without claiming exemption from physical law.
4. A synthesis of formal-verification scalability limits, abstract-interpretation domains, hierarchical verification, and assume-guarantee composition, with concrete links to domain bounding.
5. A set of falsifiable predictions and enterprise controls for energy accounting, data lineage, bounded interfaces, verification coverage, and post-deployment monitoring.

1.5 Why the Distinction Matters

Capability projections should be paired with measurements of energy per verified successful output, data provenance and drift, verification coverage, and failure-containment performance. Capital-expenditure or GDP forecasts are excluded from the technical argument because they are time-sensitive and do not establish the paper's physical or information-theoretic premises.

2. Thermodynamics of Implemented Computation

2.1 Landauer's Principle as a Strict Lower Bound

Brains and operating AI hardware are open, nonequilibrium systems. For erasing an initially unknown, unbiased bit in contact with a thermal bath at temperature T, Landauer's principle supplies the ideal lower bound [1, 2]:

$$W_{erase} \geq k_B T \ln 2$$

For a workload that irreversibly discards B_erase such bits, the corresponding idealized floor is E_min ≥ B_erase k_B T ln 2. The quantity B_erase is not equal to parameter count, FLOPs, or multiply-accumulate operations; it depends on algorithm, numerical representation, data dependencies, memory architecture, reversibility of intermediate steps, sequence length, batch size, sparsity, routing, and schedule. Equation (1) establishes that logically irreversible information loss is not thermodynamically free; it does not yield a universal scaling law in parameter count, nor does it explain the magnitude of present data-center power draw.

2.2 Practical Energy and the Operational Metric E_task

Present AI hardware operates many orders of magnitude above the Landauer floor. Measured energy is dominated by transistor switching, movement of weights and activations through memory hierarchies, inter-accelerator communication, synchronization, power conversion, cooling, redundancy, and finite-time reliability [2, 12]. The primary empirical quantity adopted in this paper is energy per accepted task outcome under a fixed quality criterion:

$$E_{task}(w) \quad \textit{(joules per verified successful task for workload } w\textit{)}$$

Reports of E_task must be accompanied by latency, throughput, hardware generation, utilization, token or sequence length, and the precise definition of success.

2.3 Claim Boundaries

Status	Defensible claim	Claim not made
Established law	Logically irreversible information loss has a nonzero minimum thermodynamic cost under stated assumptions.	Every arithmetic instruction costs k_B T ln 2, or energy is determined by parameter count.
Implementation-dependent	If a larger workload discards more logical information, its Landauer floor increases with B_erase.	A universal lower-bound curve E_min(N) follows from model size alone.
Empirical systems	Practical energy depends on hardware, memory, communication, utilization, cooling, and workload design.	The Landauer floor explains the magnitude of current data-center power use.

Table 1. Claim boundaries for thermodynamic statements.

3. Task-Relevant Representation Complexity

3.1 Information Bottleneck and Minimum Representation Rate

Shannon entropy H(X) quantifies uncertainty in an input variable but does not by itself determine the information that a predictor must retain. If X = (Y, N) and N is independent nuisance variation, H(N) can grow while the representation Z = Y remains sufficient for predicting Y [3–5]. The relevant conceptual quantity is the minimum representation information required to meet a specified prediction criterion:

$$C_Y(\varepsilon) = \inf_{q(z|x),\, \hat{y}} I(X; Z) \quad \text{subject to} \quad E[\ell(Y, \hat{y}(Z))] \le \varepsilon$$

where the infimum is taken over an explicitly chosen encoder and predictor class. C_Y(ε) is a conceptual operationalization; in the multi-objective framework of Section 5 it is treated as a conceptual component rather than an observed scalar on equal footing with measured energy or divergence.

3.2 Domain-Scope Hypothesis (Sharpened)

Hypothesis. For a fixed target error ε and a specified representation family $\mathcal{Q}$, let K be the number of distinct task-relevant decision distinctions (equivalently, the mutual information I(Y; Z*) between the multi-task label vector Y and a minimal sufficient representation Z* drawn from $\mathcal{Q}$) that the system must preserve. Then C_Y(ε; $\mathcal{Q}$) is non-decreasing in K. The independent variable is therefore the cardinality and heterogeneity of decision boundaries that must be retained at the prescribed error, not raw input entropy.

Toy formal condition. Consider a finite discrete multi-label setting in which Y = (Y_1, ldots, Y_K) with independent binary components and a deterministic encoder class of bounded description length. Under the standard rate-distortion function for independent Bernoulli sources, the minimal mutual information required to achieve average Hamming distortion ≤ ε scales linearly with K. Adding pure nuisance coordinates that are independent of all Y_i leaves the minimal rate unchanged. This supplies a concrete, falsifiable special case; the general continuous or highly correlated setting remains an empirical question.

3.3 Data Quality and Optimization Stability

Under finite capacity and imperfect optimization, task-irrelevant variation, mislabeled samples, contradictory supervision, or weakly grounded examples can degrade generalization, calibration, or convergence. Testable variables include label-noise rate, contradiction rate, out-of-distribution fraction, gradient cosine similarity, convergence variance, held-out calibration error, rare-class recall, and task performance, all measured under matched capacity, optimizer, data volume, and compute.

4. Recursive Synthetic-Data Feedback: Synthesis and Boundary Conditions

4.1 Empirical Failure Mode under Recursive Replacement

Shumailov et al. demonstrate that indiscriminate recursive training on generated data can produce model collapse: learned distributions progressively lose information about the reference distribution, with rare or tail behavior affected early [6]. The result is conditional on a recursive generation-and-replacement process; it does not imply that all synthetic data is harmful. Let P_0 be the reference data-generating distribution and P_n the distribution represented after recursion n. Define D_n = D(P_n, P_0) and T_n = TailCoverage(P_n; P_0). A collapse signature is an increasing trend in D_n, a decreasing trend in T_n, or degradation of task-specific calibration and rare-class performance.

4.2 Boundary Conditions and Mitigation

Recursive collapse is not inevitable. Retaining original data, accumulating rather than replacing samples, validating generated examples, reweighting underrepresented regions, and maintaining high-quality reference sets can materially alter outcomes [7]. Relevant experimental variables include synthetic fraction, recursion depth, replacement versus accumulation, generator quality, filtering, weighting, sample size, deduplication, and preservation of real-data anchors.

4.3 Data-Lineage Implication

The practical risk is untracked recursive exposure, not synthetic origin by itself. Training assets should retain lineage information sufficient to distinguish observed, human-authored, simulated, model-generated, transformed, and unknown data, together with model or generator version, sampling process, validation status, and intended use.

5. Multi-Objective Cost-and-Risk Framework

5.1 Why a Universal Scalar Cost Is Not Justified

Energy, representation complexity, distributional drift, verification effort, and objective conflict are expressed in different physical or statistical units. The evaluation target is therefore represented as a vector

$$M = (E_{task}, C_{rep}, D_{data}, V_{cost}, O_{conflict}, R_{operational}, \ldots)$$

where E_{task} is energy per verified successful task (measured), C_{rep} is a specified representation-complexity measure (conceptual), D_{data} is distributional drift (measured), V_{cost} summarizes verification resources and coverage, $O_{conflict}$ measures modular objective interference, and $R_{operational}$ contains application-specific reliability outcomes. Each component retains its units and uncertainty.

5.2 Application-Specific Scalarization

An organization may convert the vector into a scalar decision objective only after choosing baselines, normalization scales, and weights. Interaction terms among components are hypothesized rather than implied by the Second Law, the information-bottleneck objective, or model-collapse experiments; they must be estimated from application data if used. Some interventions reduce more than one component (e.g., domain bounding can lower both verification cost and residual risk), while others introduce trade-offs.

6. Domain-Bounded and Hierarchical Architectures

6.1 The Domain-Bounding Mechanism

Building on hierarchical control architectures and modern modular systems [8, 22, 23], we hypothesize that domain-bounded systems can achieve higher operational reliability under comparable deployment conditions because their admissible inputs, tasks, actions, and verification properties are more tightly constrained. A system scope can be written $S = (X_{adm}, T, A, P)$, where X_{adm} is the admissible input set, T the supported task set, A the permitted action set, and P the properties to be tested or verified. Bounding these sets can reduce the number of task-relevant distinctions, simplify monitoring, and make formal or empirical coverage more attainable. It does not guarantee correctness or eliminate thermodynamic, implementation, or environmental risk.

Concrete illustration. Consider a fixed ReLU network of width W and depth D used for airborne collision avoidance over a hyper-rectangle of volume V in relative state space [16]. Verification cost (solver time and memory) grows rapidly with both network size and the volume of the input region that must be certified. Restricting the certified region to the operational flight envelope and the action set to a small discrete set of advisory outputs materially reduces the number of SMT queries and the size of the reachable-set abstraction, without altering the underlying physical energy cost of inference.

6.2 Hierarchical Decomposition

Hierarchical decomposition reduces planning complexity, localizes failures, and separates decisions across spatial, logical, and temporal scales. Hierarchical reinforcement-learning frameworks use temporal abstraction and state decomposition under explicit conditions [22, 23]. A macro-policy module defines admissible goals and constraints for local optimizer nodes; each node operates over a validated input schema and bounded action space; summarized outcomes and exception signals propagate upward for policy revision.

6.3 Monitoring Statistics

A calibrated discrepancy between a module's predicted outcome distribution and the observed outcome can be scored by negative log-likelihood $s_t = -\log p_\theta(y_t \mid x_t)$ or, for a multivariate residual with predicted mean $\hat{y}_t$ and covariance Σ_t, by the squared Mahalanobis residual $(y_t - \hat{y}_t)^T \Sigma_t^{-1} (y_t - \hat{y}_t)$. Each deployment must specify the predictive model, calibration procedure, local and escalation thresholds, aggregation window, persistence criterion, severity class, and response action.

6.4 Illustrative Functional Comparison

Dimension	Biological nervous system	General-purpose AI	Domain-bounded AI
Energy constraint	Metabolic and thermal budget	Hardware, power, cooling, workload budget	Application-specific compute, latency, service budget
Input distribution	Embodied sensory stream	Broad multimodal or textual distributions	Validated schemas and bounded operating distributions
Verification scope	Behavioral and physiological testing	Broad properties and open-world behavior	Narrower input, action, and property sets
Failure containment	Biological regulation and redundancy	Architecture-dependent	Explicit isolation, authorization, monitoring, rollback

Table 2. Functional comparison of constraint mechanisms. The comparison does not claim mechanistic equivalence among biological and artificial systems.

7. Modular Objectives, Verification Scalability, and Epistemic Failure Modes

7.1 Modular Objective Conflict

In AI systems, compression, approximation, and modular optimization should be analyzed at the architectural and objective-function level. Let module i optimize $L_i(\theta)$ and let θ_s denote parameters or decisions shared with other modules. A simple

indicator of pairwise gradient conflict is $g_i^T g_j < 0$, where $g_i = \nabla_{\theta_s} L_i$. Negative alignment means that a small update improving one objective can locally worsen another. Other measures include system-level constraint violations, policy disagreement, and resource contention [13].

7.2 Universal Undecidability versus Bounded Verification

Turing proved that no general algorithm can decide whether every arbitrary program and input will halt [14]. Rice's theorem extends the limitation: every nontrivial extensional property of the partial function computed by an arbitrary program is undecidable in the general case [15]. These are universal results over unrestricted programs and inputs. They do not imply that every property of a particular fixed neural network over a bounded input domain is undecidable. The quantifiers and representation matter; Rice's theorem by itself does not prove that no deployed model can be verified.

For a fixed model, a bounded input region, and a formally specified property, verification may be decidable but computationally difficult. SMT, mixed-integer, reachability, and abstract-interpretation methods can prove properties or produce counterexamples for selected neural-network classes and domains. Reluplex, for example, verified specified properties of fixed ReLU networks used in an airborne collision-avoidance application [16]. Practical barriers include solver complexity, numerical precision, model size, state-space growth, hybrid-system interactions, incomplete environment models, and insufficiently precise properties. Verification coverage must report model version, input region, property language, solver, timeout, hardware, unresolved cases, and assumptions.

7.3 Scalability of Formal Verification

Neural-network verification (robustness, reachability, safety properties) is NP-complete in general. Non-linearities force case-splitting whose number grows exponentially with the number of unstable neurons. Early tools handled networks with a few dozen neurons; modern complete verifiers routinely handle ACAS-Xu-scale networks (approximately 300 neurons) and, with GPU-accelerated bound propagation and branch-and-bound, CIFAR-scale convolutional networks under timeouts of minutes. Frontier models with millions to billions of parameters remain out of reach for complete verification.

The principal families and their practical reach are summarized below.

Family	Representative tools	Strength	Practical limit (approx.)
SMT / LP / MIP complete	Reluplex, Marabou 2.0	Exact (sound + complete)	Hundreds to low thousands of neurons
Bound-propagation + BaB	α,β-CROWN and variants	Best empirical speed/completeness trade-off	CIFAR-scale CNNs under minute-scale timeouts
Abstract interpretation	ERAN (DeepPoly, DeepZ), AI²	Fast incomplete certification	Larger networks; precision loss grows with depth
Network reduction + backend	On-the-fly pruning / reduction	Orthogonal speed-up	Complementary; reported time reductions up to ~96%

Table 3. Principal families of neural-network verification methods and their practical reach.

Persistent bottlenecks include exponential dependence on unstable neurons and input dimensionality, accumulation of approximation error with depth, growth of cost with input-region volume, limited support for non-ReLU activations and attention mechanisms, and the absence of machine-checkable encodings for open-world properties. Domain bounding remains one of the most effective practical levers: restricting the certified input region and action set shrinks both the search tree and the required tightness of abstract bounds.

7.4 Abstract Interpretation Domains

Abstract interpretation over-approximates the set of reachable values by an abstract element drawn from a chosen domain. Sound abstract transformers for affine layers, ReLU, MaxPool and other operators guarantee that every concrete behavior is contained in the abstract result. If the final abstract element does not intersect the set of bad outputs, the property is proven. The principal domains and their precision–scalability trade-offs are as follows [25–28].

Domain	Representation	Precision	Scalability
Interval / Box	Independent bounds per neuron	Lowest	Highest
Zonotope	Center + generators (shared noise symbols)	Medium (linear dependencies)	High
DeepPoly / CROWN	Interval + two polyhedral constraints per neuron	High	High
Star sets	Affine image of an arbitrary H-polytope	Very high	Medium–low
Hybrid zonotope	Zonotope + discrete binary generators	Exact for ReLU networks	Grows with binary variables

Table 4. Principal abstract domains for neural-network verification and their precision–scalability trade-offs.

Affine layers are exact (or nearly so) for intervals, zonotopes and DeepPoly. ReLU is the dominant source of over-approximation: always-active or always-inactive phases are exact; the crossing-zero case is relaxed by a convex envelope, a new zonotope generator, or optimized linear bounds. DeepPoly/CROWN-style domains are the work-horses of many incomplete verifiers and of bound-propagation steps inside branch-and-bound; tighter abstract bounds prune more branches and reduce the search tree.

7.5 Hierarchical Verification Techniques

Hierarchical verification converts a monolithic obligation into a collection of smaller sub-problems whose solutions can be soundly composed [29–31]. Principal patterns include:

- Layer-wise geometric hierarchy. LayerCert exploits the nested hyperplane arrangement induced by successive ReLU layers, exploring activation regions hierarchically rather than as a flat polyhedral complex, thereby reducing the number and size of convex programs.
- Hierarchical safety abstract interpretation. The classical binary safe/unsafe question is replaced by a graded hierarchy of unsafe output sets; reachable sets computed by abstract interpretation are ranked against this hierarchy, yielding a more granular safety assessment at comparable or lower cost.
- CEGAR-style abstraction–refinement. A smaller over-approximating network is verified first; spurious counter-examples trigger refinement that still over-approximates the original network, producing a sound and complete procedure whose practical cost is often far lower than monolithic verification.
- Architecture-mirroring hierarchy. When the system itself is hierarchical (macro-policy plus local optimizers), verification mirrors the architecture: high-level properties are stated over abstract states; each local module is verified only with respect to its entry/exit conditions and local safety envelope.

7.6 Assume-Guarantee Composition

Assume-guarantee composition is the formal glue that links local verification obligations. A component M is verified under an assumption A about its environment and is required to provide a guarantee G about its own outputs. The triple is written ⟨A, M, G⟩. The simplest sequential rule is:

$$\langle A, M_1, G\rangle \text{ and } \langle \text{true}, M_2, A\rangle \Rightarrow \langle \text{true}, M_1 \parallel M_2, G\rangle$$

If M_1 guarantees G under assumption A, and the rest of the system (M_2) discharges A, then the composition satisfies G [32, 33].

In the neural-network setting (e.g., CoVeNN), a network N is decomposed into sequential sub-networks N = N_K ∘ ldots ∘ N_1. Intermediate predicates γ_1, ldots, γ_{K−1} are synthesized over the activation vectors that flow between sub-networks. The global property φ_in ⇒ φ_out is replaced by the chain of local obligations

$$\langle \gamma_{k-1}, N_k, \gamma_k\rangle \quad \text{for } k = 1, \text{ldots}, K$$

(with γ_0 = φ_in and γ_K = φ_out). A coarse over-approximation of the intermediate maps is computed first; if it already discharges the output property, verification succeeds. Otherwise an iterative refinement loop tightens the intermediate predicates until every local obligation is discharged or a counter-example is found. Because each sub-problem is far smaller than the original network, memory pressure on the underlying verifier is reduced; reported gains reach nearly seven times more solved properties than the same underlying verifiers used monolithically [32].

Domain-bounded hierarchical architectures supply natural contracts: the macro-policy assumes that each local module respects its typed interface and local safety envelope; each local module is verified only inside its own bounded domain; escalation and monitoring paths discharge residual assumptions at runtime. The intermediate predicates γ_k must be expressible in a form the underlying verifier can handle (linear constraints, zonotopes, abstract domains). Poorly chosen interfaces can make the composition harder rather than easier.

7.7 Open-World Specification Gaps and Expanding Failure Surfaces

Properties such as “always factually accurate,” “never harmful,” or “contextually appropriate” may lack a complete machine-checkable specification. This is distinct from theorem-level undecidability. A property cannot be exhaustively verified until its semantics, reference data, context, and acceptance conditions are formalized. Models trained on finite imperfect samples remain susceptible to spurious correlations, distribution shift, data poisoning, evasion, privacy attacks, and adversarial manipulation [18]. As system complexity increases, the number of relevant component interactions, environmental conditions, and failure pathways can grow rapidly. For deployed AI systems the more common barriers are incomplete specifications, large state spaces, solver complexity, distributed logging, environmental uncertainty, and interactions among models, tools, users, and infrastructure.

8. Enterprise Controls for Data Provenance, Drift, and Failure Containment

8.1 Provenance Is Evidence of Lineage, Not Truth

Signed manifests, dataset versioning, lineage records, and content credentials bind provenance assertions to assets and detect tampering. Cryptographic provenance indicates who or what asserted an origin and whether the associated record was altered; it does not prove that the content is truthful, accurate, representative, or fit for training [20]. Decisions should combine lineage, source reputation, validation, content analysis, intended use, and risk level.

8.2 Data Classes and Recursive-Exposure Controls

Classify assets as observed, human-authored, simulated, model-generated, transformed, or unknown. Recommended controls include:

- immutable real-data anchor sets and source-stratified holdouts;
- dataset and model versioning with signed manifests and reproducible mixture definitions;
- deduplication, contamination checks, and isolation of evaluation data from generation and tuning loops;
- tail-coverage, calibration, rare-class, and source-stratified performance tests;
- synthetic-mixture limits or review gates appropriate to the application;
- promotion criteria, rollback criteria, and post-training drift monitoring.

8.3 Typed and Validated Action Interfaces

Natural language may remain at the interaction layer, but high-impact actions should pass through typed APIs, validated intermediate representations, constrained decoding where appropriate, policy checks, authorization, confidence assessment, tool allowlists, sandboxed execution, and explicit fallback behavior for ambiguous or unparseable input [24].

8.4 Monitoring, Containment, and Recovery

Control objective	Implementation examples	Evidence / metric
Detect data drift	Source-stratified monitoring; anchor-set evaluation; tail and calibration tests	Divergence, tail recall, calibration error, source-specific performance
Prevent unsafe action	Typed interfaces; authorization; policy engine; sandbox; human approval	Unsafe-action rate, schema violations, denied actions, rollback rate
Contain failure	Module isolation; circuit breakers; rate limits; fallback models; manual takeover	Blast radius, time to containment, recovery time, repeated incident rate
Maintain accountability	Versioned logs; decision lineage; model and prompt identifiers; change control	Reproducibility rate, missing-log rate, audit coverage
Validate operation	Service-level monitoring; discrepancy thresholds; incident review	Availability, latency, success rate, anomaly persistence, unresolved incidents

Table 5. Example enterprise controls and measurable evidence. Controls must be tailored to the system, risk level, and operating context.

9. Empirical Predictions and Falsification Criteria

The following statements are hypotheses, not consequences already proved by the preceding equations. Each should be evaluated with preregistered metrics, matched controls, uncertainty intervals, and transparent reporting of negative results.

Hypothesis	Independent variable	Outcome measure	Required controls	Evidence against
Reliable-output energy rises with domain scope	Number of distinct task-relevant decision distinctions (K) at fixed ε	Joules per verified successful task; latency	Hardware generation, quality threshold, utilization, verifier	Flat or decreasing cost as K increases after matched controls
Recursive replacement reduces distributional coverage	Synthetic fraction, recursion depth, replacement vs accumulation	Tail recall, calibration, divergence from P_0	Model, sample count, filtering, real-data retention	Stable or improved coverage under replacement without anchors
Verification cost rises with property and domain scope	Input-region volume, property complexity, model size	Runtime, memory, unresolved cases, proof coverage	Verifier, precision, timeout, hardware, property language	Constant coverage and cost as scope increases
Bounded action interfaces reduce deployment failures	Typed/validated action path vs unrestricted generation	Unsafe-action rate, schema violations, rollback frequency	Same model, tasks, permissions, monitoring period	No reduction in failure rate or containment severity
Assume-guarantee composition improves solved rate	Decomposition into sub-networks + intermediate contracts	Number of properties solved; peak memory; wall-clock time	Underlying verifier, network family, property set, timeout	No gain (or degradation) relative to monolithic verification

Table 6. Proposed predictions, controls, and disconfirming evidence.

9.1 Recommended Experimental Reporting

- Report the exact model and system boundary: model weights, retrieval layer, tools, runtime policies, and verifier.
- Fix quality and reliability thresholds before measuring energy or cost.
- Separate training, adaptation, and inference energy, including infrastructure overhead where measured.
- Report data-mixture lineage, replacement or accumulation rules, and preserved reference sets.
- Use multiple drift metrics and explain why each is appropriate to the distribution and task.
- Report unresolved verification cases rather than treating timeouts as proofs or counterexamples.
- For hierarchical or assume-guarantee experiments, report the decomposition, intermediate contracts, refinement iterations, and peak memory per sub-problem.

10. Limitations and Open Research Questions

1. Landauer's limit is a foundational lower bound, not an explanation of the magnitude or observed growth of present AI energy consumption.

2. The domain-scope hypothesis is not a theorem. Expanding an input distribution with task-irrelevant variation may leave the minimal sufficient representation unchanged. The toy formal condition of Section 3.2 covers only a restricted discrete multi-label setting.
3. C_Y(ε) is not a unique off-the-shelf metric and is treated conceptually in the multi-objective vector.
4. Recursive synthetic-data degradation depends on replacement, accumulation, retention, weighting, filtering, generator quality, and finite-sample effects. Synthetic data can be useful under controlled conditions.
5. Domain bounding reduces the scope of testing and verification but does not guarantee stability, security, or correctness.
6. Turing and Rice establish universal limits for unrestricted programs; they do not prohibit bounded verification of fixed models. Practical assurance also depends on specifications, solvers, environment models, and runtime controls.
7. Abstract-interpretation domains trade precision for scalability; incomplete verification can miss counter-examples. Assume-guarantee refinement loops can themselves become expensive if intermediate contracts are poorly chosen.
8. The multi-objective vector has no universal scalarization. Organizational weights and interaction terms reflect values and application risk, not physical constants.
9. The paper does not derive a closed-form proof that general-purpose AI scaling must fail or collapse. Its contribution is a constrained, testable framework for measuring omitted observables.

11. Conclusion

Intelligent computation is physically instantiated and therefore operates under finite energy, time, memory, communication, and reliability constraints. Landauer's principle establishes a lower bound on logically irreversible information loss, while practical AI energy is dominated by implementation, data movement, communication, cooling, and finite-time reliability. The operational quantity of primary interest is energy per verified successful task under a fixed quality criterion.

Information-bottleneck concepts support a more careful representation claim than a raw-entropy argument: the relevant quantity is the information needed to preserve task-relevant distinctions at a specified error level. That quantity is hypothesized to grow with the number of distinct decision distinctions that must be retained, but nuisance entropy can be discarded and the relationship must be tested rather than assumed.

Recursive training on generated data can degrade distributional coverage under replacement and insufficient preservation of the reference distribution. The effect is conditional, not universal. Data lineage, retained real-data anchors, validation, reweighting, and monitoring are therefore more defensible controls than a blanket rejection of synthetic data.

Domain-bounded execution, hierarchical decomposition, typed action interfaces, modular isolation, and bounded verification can reduce the scope over which failures must be detected and controlled. Abstract-interpretation domains supply the geometric language of incomplete verifiers; branch-and-bound and cutting-plane methods extend reach; hierarchical and assume-guarantee composition convert monolithic obligations into chains of smaller, contract-linked sub-problems. These techniques reduce risk by construction but do not eliminate residual failure, undecidability in the unrestricted case, or specification gaps in open-world behavior.

The paper's central falsifiable claim is therefore: for a fixed hardware generation and target task-success threshold, broadening the admissible task and input scope (measured by the number of distinct task-relevant decision distinctions that must be preserved) is hypothesized to increase energy per verified successful output and the cost of achieving specified verification coverage. Under recursive replacement of reference data with unvalidated generated samples, distributional tail coverage is hypothesized to decline with recursion depth. These effects should weaken when task-relevant complexity is held constant, original data are retained, generated samples are validated and reweighted, or the system is decomposed into independently testable bounded modules whose local obligations are linked by assume-guarantee contracts.

The practical implication is not that AI investment or research should stop. It is that capability forecasts and system evaluations should include energy per verified successful output, data lineage and drift, verification coverage, objective conflict, and failure-containment performance. Treating benchmark loss as a complete proxy for reliable and economical capability leaves important physical and systems-level observables unmeasured.

References

[1] R. Landauer, "Irreversibility and heat generation in the computing process," IBM Journal of Research and Development, vol. 5, no. 3, pp. 183–191, 1961.

[2] C. H. Bennett, "The thermodynamics of computation—a review," International Journal of Theoretical Physics, vol. 21, no. 12, pp. 905–940, 1982.

[3] C. E. Shannon, "A mathematical theory of communication," Bell System Technical Journal, vol. 27, no. 3, pp. 379–423, 1948.

[4] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in Proc. 37th Allerton Conference on Communication, Control and Computing, 1999.

[5] A. A. Alemi et al., "Deep variational information bottleneck," in International Conference on Learning Representations, 2017.

[6] I. Shumailov et al., "AI models collapse when trained on recursively generated data," Nature, vol. 631, pp. 755–759, 2024.

[7] M. Gerstgrasser et al., “Is model collapse inevitable? Breaking the curse of recursion by accumulating real and synthetic data,” arXiv:2404.01413, 2024.

[8] M. Moganti et al., “Network intelligence: A systems architecture perspective,” IEEE Communications Magazine, 2001 (and subsequent hierarchical-control extensions).

[9] M. E. Raichle and D. A. Gusnard, “Appraising the brain’s energy budget,” Proceedings of the National Academy of Sciences, vol. 99, no. 16, pp. 10237–10239, 2002.

[10] J. Kaplan et al., “Scaling laws for neural language models,” arXiv:2001.08361, 2020.

[11] J. Hoffmann et al., “Training compute-optimal large language models,” in Advances in Neural Information Processing Systems, 2022.

[12] D. Patterson et al., “Carbon emissions and large neural network training,” arXiv:2104.10350, 2021.

[13] T. Yu et al., “Gradient surgery for multi-task learning,” in Advances in Neural Information Processing Systems, 2020.

[14] A. M. Turing, “On computable numbers, with an application to the Entscheidungsproblem,” Proceedings of the London Mathematical Society, vol. s2-42, no. 1, pp. 230–265, 1937.

[15] H. G. Rice, “Classes of recursively enumerable sets and their decision problems,” Transactions of the American Mathematical Society, vol. 74, no. 2, pp. 358–366, 1953.

[16] G. Katz et al., “Reluplex: An efficient SMT solver for verifying deep neural networks,” in Computer Aided Verification, Springer, 2017.

[17] D. H. Wolpert, “Physical limits of inference,” Physica D, vol. 237, no. 9, pp. 1257–1281, 2008.

[18] I. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in International Conference on Learning Representations, 2015.

[19] National Institute of Standards and Technology, “Artificial Intelligence Risk Management Framework (AI RMF 1.0),” NIST AI 100-1, 2023.

[20] Coalition for Content Provenance and Authenticity, “C2PA Technical Specification,” 2024.

[21] European Commission, “Proposal for a Regulation laying down harmonised rules on artificial intelligence (AI Act),” 2021/0106(COD).

[22] R. S. Sutton, D. Precup, and S. Singh, “Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning,” Artificial Intelligence, vol. 112, no. 1–2, pp. 181–211, 1999.

[23] P. Dayan and G. E. Hinton, “Feudal reinforcement learning,” in Advances in Neural Information Processing Systems, 1993.

[24] J. H. Saltzer and M. D. Schroeder, “The protection of information in computer systems,” Proceedings of the IEEE, vol. 63, no. 9, pp. 1278–1308, 1975.

[25] G. Singh et al., “An abstract domain for certifying neural networks,” Proceedings of the ACM on Programming Languages, vol. 3, no. POPL, 2019.

[26] H. Zhang et al., “Efficient neural network robustness certification with general activation functions,” in Advances in Neural Information Processing Systems, 2018 (CROWN).

[27] S. Wang et al., “Beta-CROWN: Efficient bound propagation with per-neuron split constraints for neural network robustness verification,” in Advances in Neural Information Processing Systems, 2021.

[28] T. Gehr et al., “AI2: Safety and robustness certification of neural networks with abstract interpretation,” in IEEE Symposium on Security and Privacy, 2018.

[29] C. H. Lim, R. Urtasun, and E. Yumer, “Hierarchical verification for adversarial robustness,” arXiv:2007.11826, 2020.

[30] L. Marzari, I. Mastroeni, and A. Farinelli, “Advancing neural network verification through hierarchical safety abstract interpretation,” arXiv:2505.05235, 2025.

[31] G. Katz et al., “The Marabou framework for verification and analysis of deep neural networks,” in Computer Aided Verification, 2019 (and Marabou 2.0).

[32] H. Duong, D. Shriver, T. Nguyen, and M. Dwyer, “Compositional neural network verification via assume-guarantee reasoning,” in Advances in Neural Information Processing Systems, 2025 (CoVeNN).

[33] C. S. Păsăreanu, D. Gopinath, and H. Yu, “Compositional verification for autonomous systems with deep learning components,” white paper, NASA Ames / Boeing, 2018.