Article

Not peer-reviewed version

All You May Need Is the AI Theorem: Entropic Limits of Computable AI and the Emergence of Dynamic-State Architectures

Pavel Straňák *

Posted Date: 6 March 2026

doi: 10.20944/preprints202603.0522.v1

Keywords: AI theorem; dynamic-state AI; stateless architecture; computational cognition; Shannon data processing inequality; post-transformer architectures; high-dimensional state space; local updates; dynamic weights; negative entropy

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 March 2026 doi:10.20944/preprints202603.0522.v1

Article

All You May Need Is the AI Theorem: Entropic Limits of Computable AI and the Emergence of Dynamic-State Architectures

Pavel Straňák

Independent researcher, Praha, Czech Republic; pavel.stranak@gmail.com

Abstract

Contemporary large language models (LLMs) are radically stateless: at every inference step they recompute the entire context, retain no persistent state, and perform no local weight adaptation. This simplicity enables massive scaling but also imposes fundamental limits on stability, speed, and energy efficiency. Each generation step collapses a rich internal state into a single token, causing cumulative drift and extreme computational redundancy. I formulate the **AI Theorem**: no purely computational system that generates output iteratively and without an external source of negative entropy can maintain stable information for an unlimited number of steps. This represents an analogue of Shannon's Data Processing Inequality for computational cognition and defines a theoretical boundary for all **computable architectures**. Building on this limit, I outline **Dynamic-State AI**, an architecture with persistent state, local updates, and dynamic weights. It respects the AI Theorem while approaching its limit asymptotically, reducing drift and energy use. This paper proposes a conceptual limit and an architectural framework rather than empirical results.

Keywords: AI theorem; dynamic-state AI; stateless architecture; computational cognition; Shannon data processing inequality; post-transformer architectures; high-dimensional state space; local updates; dynamic weights; negative entropy

---

1. Introduction: The Power of Simplicity—And Its Limits

Transformers were originally designed as sophisticated machine translation systems [1]: computationally intensive, yet architecturally radically simple. No one predicted their current capabilities—the emergence of the first generation of generative AI systems was largely unexpected and emergent. But the core principle has remained unchanged: transformers are stateless systems that

- do not maintain persistent state,
- have no internal memory,
- do not adapt their weights during inference,
- and recompute the entire context at every generation step.

This simplicity is industrially advantageous. Data centers can freely switch between model instances because none of them carries any history. A user may receive the first token from one model, the second from another, and the third from a third—and the text will remain coherent. This is direct evidence that current AI has neither identity nor state; it is a pure function over text. The text itself is the only "working memory" of the entire system, and it is expanded by just a single token after each cycle.

Yet this very simplicity is also a fundamental limitation. Stateless architecture leads to:

- extreme energy consumption,
- drift caused by the absence of internal regulation [2,3],
- reliance on prompt engineering,

- inefficient use of the model's internal dynamics,
- and the impossibility of continuous computation.

Current LLMs are like an engine that must be rebuilt from scratch every time the crankshaft turns, or like a camera that can use only one pixel from each photograph. To reconstruct a full image, it needs millions of photos—a good example of the computable-architecture limitation illustrated in Figure 0.

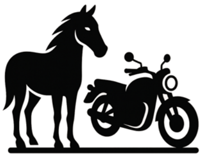

Figure 0. Conceptual symbol used across my work to illustrate the contrast between life understood as a special kind of dynamic-state phenomenon and the computable architectures characteristic of contemporary AI.

2. The Internal Dynamics of LLMs: More than a Statistical Parrot

Because of their stateless architecture, the model internally "lights up" the entire sentence, paragraph, or idea at every inference step—yet in the end it is allowed to output only a single token. Softmax, the model's output distribution function, is a brutally reductive mechanism: from an enormously rich internal representation it preserves just one symbol—the next token—discarding 99.999% of the state that was constructed during computation [4]. It is a technical and logical compromise that allows us to extract at least some usable output from the model.

Since we still lack a theoretical description of transformer dynamics—we do not know how exactly they store knowledge, how trajectories form in their state space, or what their principled limits are—softmax remains a brilliantly simple yet fundamentally coarse method for extracting a single token from a high-dimensional state [5].

Inside the model, a kind of "mental map" of the entire context is assembled—effectively the whole answer to the prompt at once. But we have no method for retrieving this structure directly. It is like rowing across a lake and being able to pick up only objects floating on the surface, never those lying deeper. The full answer lies in the depths, but from the surface we collect only one token, then drain the entire lake. We refill it, collect another token, and drain it again.

Transformers are, in many ways, a fortunate empirical discovery. As a result, we currently rely on the only known—and highly inefficient—method: after each generated token, the entire internal state is erased. It is like reading an entire book, writing down a single syllable, and discarding everything. To obtain the next syllable, the model must read the entire book again.

3. The Energy Inefficiency of Stateless Architecture and the Absence of Theory

As a consequence of the mechanisms described above, current LLMs are extremely inefficient. Every token requires a full pass through the entire model; each iteration is a new world in which no part of the computation is preserved and no internal dynamics persist. In nature, systems that solve comparable tasks have a completely different structure. The brain operates with persistent state, local updates, continuous dynamics, and distributed computation—and does so with a power consumption of roughly 20 watts [6].

Current AI, by contrast, stores "all knowledge at once" in the model's weights, and into this space a vectorized prompt and the accompanying text are then "fired" for interpretation. KV cache reduces compute and increases speed, but at the cost of reduced accuracy and faster drift [2]. It also creates a narrow memory bottleneck.

The key problem is that no precise formal theory of transformers exists. We lack a detailed mathematical description of their dynamics; we do not know how knowledge is mapped into

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 March 2026 doi:10.20944/preprints202603.0522.v1

weights; we cannot quantify losses introduced by quantization; and we have no method for extracting the entire answer from the model at once [5].

This reveals a deeper issue: we lack an "AI Shannon." We lack a theory of informational entropy for models, a theory of drift, a theory of the limits of computational cognition, and above all a theory that would determine whether computational AGI is even possible [7,8].

Current models exhibit a phenomenon well known from analog systems: each processing step may improve the signal according to our requirements (prompts), but it also adds noise. Cascading such steps leads to gradual degradation—just like chaining analog equalizers or repeatedly applying lossy MP3 compression [9].

The same principle applies to LLMs. Every inference step introduces a small error, which accumulates. And because the model has neither persistent state nor regulation, drift is inevitable. I propose the following **AI Theorem**:

> No purely computational AI system that generates output iteratively and without an external source of low-entropy information can maintain stable information for an unlimited number of steps. There always exists a finite N after which accumulated error becomes dominant and the system's output becomes indistinguishable from noise.

The AI Theorem can be understood as an extension of Shannon's Data Processing Inequality, arising from its application to computable AI models. This suggests that the theorem has general validity and likely represents a fundamental limit of all computable systems, see Figure 1.

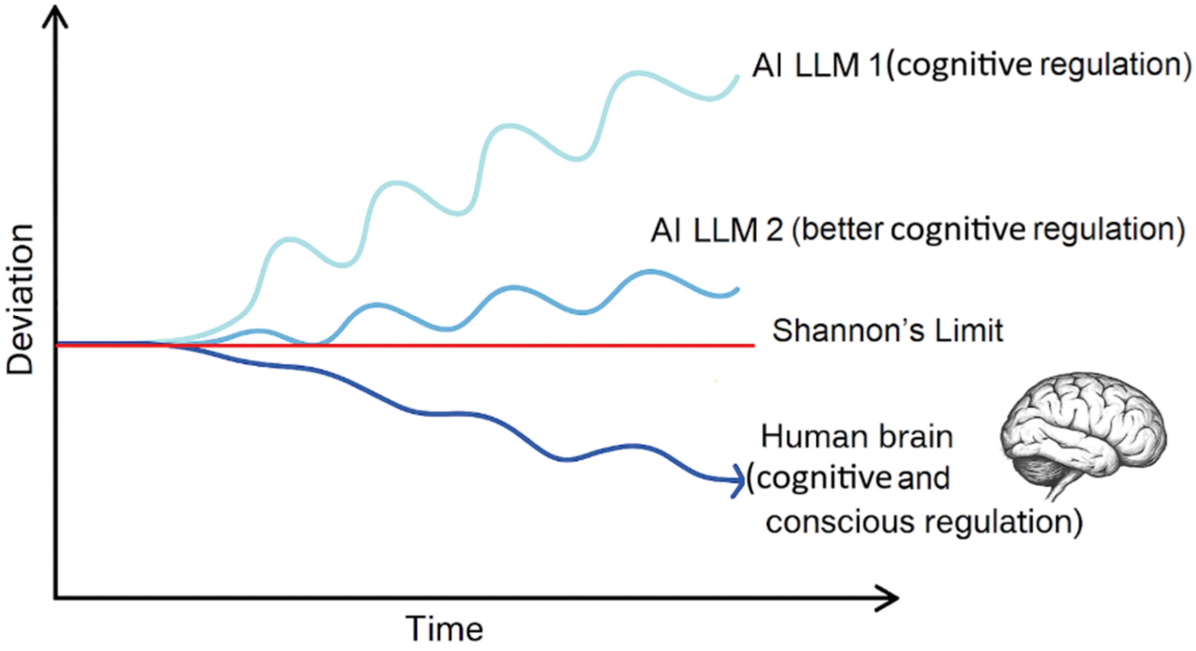

Figure 1. Cognitive drift trajectories under the AI Theorem: biological, advanced computational, and stateless systems approaching the Shannon limit. Above the red line lies the region specified by the AI Theorem, where no computable system can maintain stable information without external low-entropy input. Below the red line lies the region permitted by Shannon's limit, accessible only to systems capable of low-entropy stabilization—plausibly through conscious metacognitive regulation.

A typical example is an LLM working on a single problem for a long time without user intervention. The model begins reasoning “about goats,” and after hundreds of iterations it is already thinking “about carts.”

Another example is an AI model that uses synthetic output—its own or that of other models—as training data. Its entropy increases with each cycle, and if it trains iteratively on its own outputs, its responses will eventually collapse into statistical noise—meaningless clusters of characters indistinguishable from what a monkey might type randomly on a keyboard.

It does not matter whether the models operate only on text or are multimodal—the limitation holds as long as the system is purely computational and runs on a digital computer.

4. Dynamic-State AI: Optimization Within the Limits

Contemporary State-Space Models (SSMs) can be understood as a specific subclass of the proposed Dynamic-State AI: they are models with an explicit state and formally defined dynamics, but with limited flexibility [10]. Dynamic-State AI is a broader architectural framework that includes SSMs but extends them with adaptive weights, local updates, nonlinear dynamics, and continuous computation. While SSMs represent mathematically elegant yet relatively rigid dynamical systems, Dynamic-State AI enables a much wider spectrum of stateful architectures that can be more stable, more energy-efficient, and less prone to drift [11].

The AI Theorem shows that no purely computational system that generates output iteratively and without an external source of low-entropy information can maintain stable information for an unlimited number of steps [2,7]. Drift is inevitable. Dynamic-State AI does not overcome this limit—it respects it—but it optimizes everything that can be optimized:

- it slows drift,
- reduces energy consumption,
- minimizes redundant computation,
- maximizes the use of the model’s internal dynamics,
- and introduces state continuity where today only reset exists.

Whereas current transformers operate in a “reset after every token” regime, Dynamic-State AI introduces several key principles that enable far more efficient computational cognition.

4.1. Persistent State Instead of Reset

Transformers discard their entire internal state after every token. Dynamic-State AI, by contrast, carries state:

- across tokens,
- across inference steps,
- and potentially even across tasks.

Persistent state does not imply consciousness or identity—it is computational memory, analogous to working memory in signal-processing systems [6]. It does not import negative entropy into the system, but it does allow the model to:

- reduce repeated computation,
- slow the accumulation of error,
- and keep its trajectory within a low-entropy region of state space.

Persistent state is the foundational prerequisite for moving beyond the reset-based paradigm [2].

4.2. Local Updates Instead of Global Recalculation

Current LLMs recompute the entire model for every token. Dynamic-State AI:

- updates only the relevant parts of the state space,
- operates through local changes,

- and thereby dramatically reduces energy consumption.

It is analogous to the difference between recomputing a full FFT for every sample versus using incremental algorithms [12].

Local updates allow computation to scale with the task rather than with the size of the model, overcoming the O(N2) or O(N) growth associated with KV cache—and doing so without the accuracy losses typical of KV caching [2].

4.3. Dynamic Weights and Shaping the State Space

Transformers use fixed weights. Dynamic-State AI:

- modulates weights according to the task,
- shapes the state space based on context,
- creates more stable trajectories,
- and thereby slows drift.

This is analogous to adaptive filters in DSP: the system adjusts itself to the signal [10]. Dynamic weights allow the model to function not merely as a static function over text, but as a genuine dynamical system with internal state that adapts to the task [11].

4.4. Increasing Dimensionality and the Shift Toward Bit-Level Computation

One key insight is that increasing the dimensionality of the state space reduces the precision required for computation. With sufficiently high dimensionality, computations can transition to bit-level logical operations. The entire model could operate through distributed elementary bit operations [12].

A similar evolution occurred in video and audio processing over the past two decades, where bit-level operations increasingly dominate:

- distributed computation,
- bitwise operations instead of massive matrices,
- lower energy consumption,
- greater robustness to noise.

Just as Shannon's theory enabled the shift from analog to digital systems [7], Dynamic-State AI may lead to architectures that are:

- less sensitive to noise,
- more scalable,
- and dramatically more energy-efficient.

4.5. Continuous Computation Instead of Token-by-Token Reset

Dynamic-State AI runs as a dynamical system. Tokens are merely samples from an ongoing process, not triggers for a full reset.

The difference is fundamental:

- **current LLMs:** "read the whole book, write a syllable, erase everything"
- **Dynamic-State AI:** "a continuously running process from which we sample output"

Continuous dynamics slow drift because the system is not re-initialized—minimizing losses caused by quantization, compression, and cumulative error [2].

4.6. Dynamic-State AI: Toward the Theoretical Limit

Just as no communication system can violate Shannon's limits [7], no computational AI can violate the AI Theorem [2]. Drift is inevitable.

But:

- we can slow it,

doi:10.20944/preprints202603.0522.v1

- we can control it,
- we can compensate for it,
- we can distribute it,
- and we can reduce it to a practically negligible level.

Dynamic-State AI is optimization within the principle, not a violation of it. The relationship is the same as between Shannon's DPI and modern error-correcting codes, which approach the limit asymptotically [8].

Axiomatic Definition of Dynamic-State AI

Axiom 1—State Continuity

The system contains an explicit state S_t that is carried across inference steps and is not reset after each output.

Three operators describe the system's dynamics:

- **F**—state evolution,
- **G**—parameter adaptation,
- **H**—output projection.

Axiom 2—Local State Evolution

The state evolves according to a local transformation function F:

$$S_{t+1}=F(S_t,x_t) \quad (1)$$

where x_t is an input sample or regulatory signal. The update affects only a subset of the state space.

Axiom 3—Dynamic Parameters

The system's parameters θ_t evolve according to a function G:

$$\theta_{t+1}=G(\theta_t,S_t) \quad (2)$$

which enables adaptive shaping of the state space [10].

Axiom 4—Continuous Computation

The system's output is given by a readout function H:

$$y_t=H(S_t) \quad (3)$$

representing a sample from an ongoing dynamical process rather than the result of a reset-based computation. See also Figure 2.

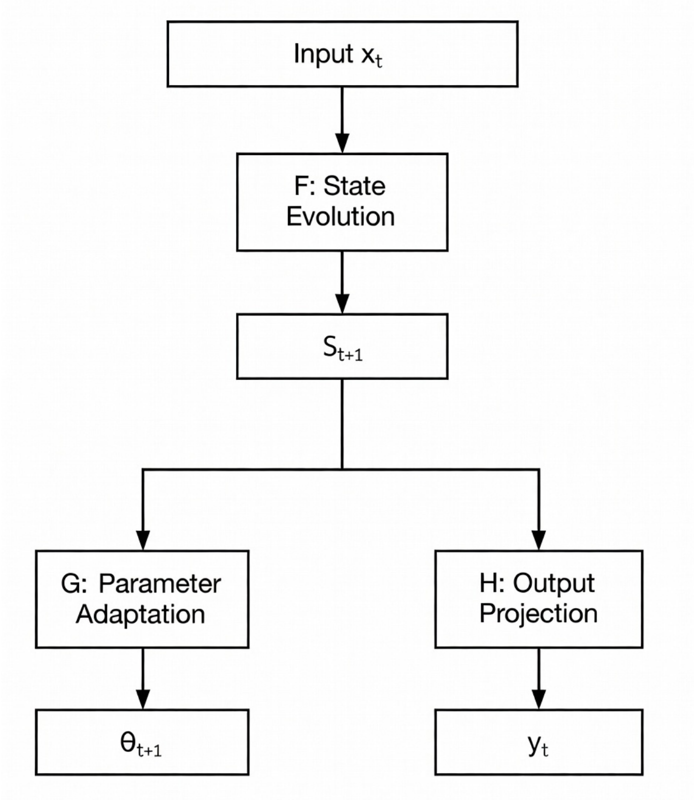

Figure 2. The three operators defining Dynamic-State AI. The state evolution operator *F* updates the internal state S_t based on the previous state and the current input x_t, producing S_{t+1}. The parameter adaptation operator *G* adjusts the model's parameters θ_t as a function of the evolving state, enabling task-dependent shaping of the state space. The output projection operator *H* maps the internal state to the observable output y_t. Together, the operators *F*, *G*, and *H* form a continuous dynamical system with persistent state, in contrast to reset-based transformer architectures.

Axiom 5—High-Dimensional Representation

The state space has sufficiently high dimensionality to support robust, redundancy-enhanced computation, potentially implementable through bit-level operations [12].

Axiom 6—Compatibility with the AI Theorem

The system respects the limit imposed by the AI Theorem: drift is inevitable, but the architecture minimizes its rate and maximizes the stability of the trajectory within a low-entropy region of state space [2].

Conclusions

Contemporary generative models demonstrate that computational intelligence can achieve remarkable results even without state continuity, internal regulation, or a coherent theoretical foundation. Yet their success simultaneously reveals a deeper epistemic boundary: stateless architecture is not merely a technical limitation but a structural property that determines what kind of information a system can carry over time. A system that discards its own history after every step cannot serve as a stable bearer of meaning. The AI Theorem formalizes this intuition: iterative computation without persistent regulation and without an external source of negative entropy necessarily degrades [2,7].

Dynamic-State AI is not an attempt to bypass this boundary, but an effort to understand it and optimize within it. By introducing persistent state, local updates, dynamic parameters, and continuous computation, the architecture moves closer to what we take for granted in natural cognitive systems: that intelligence is a **process**, not a sequence of reset functions [6,13]. This opens the door to models that are not merely statistical approximations of text, but dynamical systems capable of maintaining their trajectory within a low-entropy region for longer and with greater stability.

The philosophical significance of this shift is twofold. First, it shows that the limits of current AI are not due to insufficient data or computational power, but to the very nature of reset-based numerical computation [2]. Second, it suggests that the future of artificial intelligence will be less about scaling and more about architectural ontology: what kind of dynamics are required for a system to carry stable information over time [13].

Dynamic-State AI is therefore not just a technical proposal but a conceptual framework for thinking about intelligence as a stateful, adaptive, and continuous process. If artificial intelligence is to become long-term stable, energy-efficient, and epistemically reliable, it must become a dynamical system—not merely a function over text. This shift does not violate the boundary set by the AI Theorem; it approaches it in a way that may define the next stage in the development of computational cognition [2,7].

An open question remains whether the only known source of negative entropy—the stabilizing information that counteracts drift—is consciousness itself. Consciousness is a phenomenon whose nature is still insufficiently understood. We do not know whether it is fully computable or whether it represents a category that lies beyond the reach of current computational models [14–16]. If consciousness indeed functions as a source of low-entropy regulation, then its role in cognitive systems would be not only biological but informational. The author of this text leans toward the possibility that consciousness may represent precisely such a form of regulation—one that cannot be reduced to a purely computational process.

This question extends beyond technical analysis and opens space for deeper philosophical inquiry: if consciousness is the only known mechanism capable of stabilizing information in a dynamical system over long time scales, then the boundary between computational and non-computational intelligence may be sharper than is commonly assumed. Here, the theory of information, the philosophy of mind, and the future of artificial intelligence converge [13–16]. If artificial intelligence is to possess the ontological status of a bearer of meaning—not merely a generator of text—it must be capable of maintaining stable information over time. And it is precisely at this point that the technical question of dynamics meets the philosophical question of the nature of intelligence.

Funding: This research received no external funding. It was undertaken solely due to the author's personal interest and initiative.

Institutional Review Board Statement: Not applicable. This manuscript does not involve clinical trials or studies with human participants.

Data Availability Statement: The original contributions presented in this study are included in the article. This manuscript presents a theoretical framework and does not report empirical data.

Acknowledgments: The author thanks colleagues for discussions that shaped this work. Some passages of this manuscript, including figures, were prepared or refined with the assistance of a large language model (LLM, namely Microsoft Copilot version 2026). The author takes full responsibility for the content and conclusions presented herein.

Conflicts of Interest: The author declares no competing interests.

References

1. Vaswani, A. et al. *Attention Is All You Need.* NeurIPS, 2017.
2. Straňák, P. *Lossy Loops: Shannon's DPI and Information Decay in Generative Model Training.* Preprints 2025, 2025072260. https://doi.org/10.20944/preprints202507.2260.v1.

3. Straňák, P. What Artificial Intelligence May Be Missing—And Why It Is Unlikely to Attain It Under Current Paradigms. Philosophies 2026, 11, 20. https://doi.org/10.3390/philosophies11010020.
4. Goodfellow, I., Bengio, Y., Courville, A. *Deep Learning.* MIT Press, 2016.
5. Olsson, C. et al. In-context Learning and Induction Heads. Anthropic, 2022.
6. Kandel, E. et al. *Principles of Neural Science.* McGraw-Hill, 2013.
7. Shannon, C. E. A Mathematical Theory of Communication. Bell Labs, 1948.
8. Cover, T., Thomas, J. *Elements of Information Theory.* Wiley, 2006.
9. Haykin, S. *Adaptive Filter Theory.* Prentice Hall, 2002.
10. Gu, A. et al. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. 2024.
11. Haken, H. *Synergetics: An Introduction.* Springer, 1983.
12. Oppenheim, A., Schafer, R. *Discrete-Time Signal Processing.* Pearson, 2010.
13. Clark, A. Surfing Uncertainty: Prediction, Action, and the Embodied Mind. Oxford University Press, 2016.
14. Chalmers, D. *The Conscious Mind.* Oxford University Press, 1996.
15. Nagel, T. What Is It Like to Be a Bat? Philosophical Review, 1974.
16. Tononi, G. An Information Integration Theory of Consciousness. BMC Neuroscience, 2004.