Research Article

A Non-Ergodic Framework for Understanding Emergent Capabilities in Large Language Models

Javier Marin[1]

[1]. Independent researcher

Large language models have emergent capabilities that come unexpectedly at scale, but we need a theoretical framework to explain why and how they emerge. We prove that language models are actually non-ergodic systems while providing a mathematical framework based on Stuart Kauffman's theory of the adjacent possible (TAP) to explain capability emergence. Our resource-constrained TAP equation demonstrates how architectural, training, and contextual constraints interact to shape model capabilities through phase transitions in semantic space. We prove through experiments with three different language models that capacities emerge through discrete transitions guided by constraint interactions and path-dependent exploration. This framework provides a theoretical basis for understanding emergence in language models and guides the development of architectures that can guide capability emergence.

Corresponding author: Javier Marín, javier@jmarin.info

1. Introduction

Current research in large language models has unveiled increasingly complex capabilities that emerge unpredictably at scale. These emergent capabilities, from complex reasoning to zero-shot learning, appear without explicit training[1]. While observing these phenomena, we found patterns suggesting fundamental similarities with complex biological systems: capabilities emerged through sudden transitions rather than gradual improvements[2], model behavior showed strong dependence on context history[3], and next-token predictions varied significantly based on the path taken to reach a particular state[4]. These observations suggested that language models, like biological systems, might be fundamentally non-ergodic in nature. This insight led us to explore theoretical biology frameworks, particularly Stuart Kauffman's theory of adjacent possible (TAP), which describes how biological systems navigate their possibility spaces through restricted exploration[5]. Similar to how a cell in a developing organism navigates a limited array of possibilities influenced by its environment and history rather than choosing its next state at random, a language model's next-token prediction arises from a complex interaction of collected patterns and existing constraints. This paper presents both theoretical proof of language models' non-ergodicity and a novel framework based on TAP that explains their emergent capabilities.

1.1. Current frameworks for emergent capabilities in language models

Actual advances in AI systems research primarily emphasize empirical observations of increases in capability[6][2] and scaling laws[7], yet we lack a unified theoretical framework to explain the underlying mechanisms.

Some studies approaching this challenge have shown the stochastic nature of these emergent properties. Research on next-token prediction (NTP) proves that these capabilities arise from intrinsically stochastic processes[8], challenging deterministic interpretations of model behavior. Analyses of probability spaces in language models, including softmax distributions[9], entropy in token predictions[9][10], and temperature sampling effects[11], suggest that this emergence follows complex probabilistic patterns that current frameworks struggle to explain.

An important restriction in evaluating these emergent capabilities is the underlying assumption of ergodicity in current approaches[12][13]. This assumption breaks down when observing how capabilities emerge through path-dependent processes. In language models, each new capability depends on the assembled context and previous token sequences, leading to different probability distributions for the same token in different contexts. This context-dependent behavior contravenes fundamental ergodic principles, which require time and ensemble averages to be equivalent[14]. The non-linear dynamics seen in language models[15] create a kind of “memory system” through the context window. This leads to path dependence and temporal asymmetry, which are qualities of non-ergodic systems[16]. This fundamental feature of language models suggests that a framework that explicitly accounts for the historical path by which capabilities emerge is necessary to describe emergence rather than relying on simple averages of states.

1.2. Motivation for a biological-inspired approach

One of the most important differences between research in physics or disciplines such as theoretical biology and research in mathematics or computer science is that theoretical physicists tend to ignore everything that they consider irrelevant, focusing only on the fundamentals. When trying to describe a phenomenon, physicists do not take into account the observable details but rather try to find the fundamental laws that underlie it. This is why fundamental laws, such as conservation laws, symmetries, or phase transitions, are present in different fields, such as astrophysics, general relativity, particle physics, or quantum mechanics[17]. In this research, we aim to identify some fundamental laws that govern AI systems. To do this, we will apply some of the current fundamental laws of physics and biology to artificial intelligent systems, particularly large language models.

Similar to how a cell in a developing organism navigates a limited array of possibilities influenced by its environment and history rather than choosing its next state at random[5], a language model's next-token prediction arises from a complex interaction of collected patterns and contextual constraints. This idea suggests that some theoretical biology frameworks, particularly Stuart Kauffman's theory of adjacent possible or TAP[5][18], might offer a valid framework to understand how language models navigate their possibility spaces when predicting the next token. In biological systems, evolution is considered a process evolving within a set of constrained possibilities and random events. This evolution creates, without selection acting to do so, new "adjacent possible empty niches" which enable new possible directions of evolution. These paths are accessible based on the system's current state and guide future evolution by removing incompatible random explorations through selective exclusion[19]. Longo & Montévil[20] further develop this understanding through their theory of "extended critical transitions," arguing that biological systems perpetually operate in a critical state, continuously redefining their own phase space. They emphasize that biological systems don't just explore pre-existing possibility spaces, but actively create new dimensions of these spaces through their evolution[21]. This idea connects with how language models generate the next token, where each possible selection not only explores but also reshapes the space of possible continuations.

This biological approach to systems organization and adaptation has interesting connections with current artificial intelligence architectures. LeCun's H-JEPA framework -Hierarchical Joint Embedding Predictive Architecture-[22] establishes a framework for autonomous intelligence that shows many similarities with the adaptive mechanisms of biological systems. H-JEPA builds hierarchical world models via prediction-based learning, just like how living organisms form and maintain their organizational structure. H-JEPA's emphasis on learning world models through prediction is consistent with theoretical biologist Stuart Kauffman's theory of how living systems explore their possibility spaces via restricted exploration[5].

Novel research in AI systems has further strengthened these connections between biological systems and AI systems. For example, Zador[23] provides relevant insights into how biological neural networks' efficiency and sparsity could inform artificial systems design. Richards et al.[24] explain how hierarchical learning in deep networks parallels biological development, showing similar behaviors in how both systems build increasingly complex representations. This biological inspiration extends to architecture design, showing how principles from neuroscience can guide the development of more adaptive and efficient artificial neural networks[25]. These findings illustrate how, while modern AI architectures are still far from biological complexity, they are beginning to reflect biological system behavior.

Another relevant research area in AI systems is continual learning, which clearly draws parallels between the ability of biological and artificial systems to acquire new knowledge while retaining existing capabilities[26]. Research on catastrophic forgetting and solutions inspired by biological memory consolidation also provides valuable insights into how systems can maintain and expand their possibility spaces over time[27]. Researchers have further shown how artificial systems can achieve the kind of open-ended learning observed in biological systems[28]. The biological perspective is likewise enriched by the work of Van de Ven & Tolias[29], providing a theoretical framework for identifying diverse forms of continuous learning that closely resemble biological adaptation processes.

The intersection of these areas, from biological self-organization to modern AI architectures, suggests a deeper connection between how biological and artificial systems navigate their possibility spaces. This connection becomes clearer when we consider how language models explore their semantic spaces by combining learned patterns with contextual constraints in ways that are similar to a living system's restricted creativity.

1.3. Empirical evidence in LLMs

Recent empirical research provides compelling evidence for how language models navigate their possibility spaces in ways that mirror biological systems' constrained exploration. Recent work presented at NeurIPS 2024 on in-context exploration in large language models demonstrates how these systems' exploration capabilities are fundamentally shaped by constraints similar to those observed in biological systems[30]. The finding that models need clear exploration hints and external memory support to exhibit strong exploration behavior fits with Kauffman's theory of how systems navigate their adjacent possible spaces[5].

The poster presented at NeurIPS 2024 by A. Krishnamurthy, K. Harris, D. J Foster, C. Zhang, and A. Slivkins makes a relevant observation of "suffix failures," where models fail to explore optimal choices even after initial exposure. This evidence suggests that, like in complex biological systems, language models operate within constrained possibility spaces where exploration is limited by both architectural and contextual factors. The need for external history evaluation corresponds with how biological systems need ambient framing to increase their exploration ability. These empirical results support our idea by proving how different constraints in language models interact to shape the exploration of the next token space—akin to the interacting constraints in biological systems that shape possibility spaces[31]. Finally, the identified exploration failures provide evidence for the non-ergodic nature of these systems[32], where past trajectories fundamentally influence future exploration capabilities. The convergence between theoretical predictions and empirical observations strengthens our intention to apply complex biological systems frameworks to analyze language model behavior.

2. Probabilistic spaces in language models

The foundations of probabilistic modeling in language trace back to Shannon's information theory[33] and early statistical NLP[34]. This framework initially treated language as a stochastic process where words could be predicted based on their statistical co-occurrence patterns. Modern language models have progressed past traditional static statistical methods, using dynamic, context-sensitive probability distributions via transformer architectures[35]. This evolution represents a fundamental shift from considering language as an essentially statistical phenomenon to viewing it as a dynamic, context-dependent system.

Current transformer-based models provide probability distributions that differ fundamentally from traditional statistical language models in several aspects, like contextual dependency or sampling dynamics. Probability distributions are computed through complex attention mechanisms[35], capturing semantic uncertainty and ambiguity[36]. The formation process involves complex interactions between attention heads[37]. Beyond basic temperature sampling, actual innovations include the nucleus sampling scheme, top-p sampling[9], and top-k sampling[38]. Meister & Cotterell[11] observed that models learn "only a subset of the tendencies" rather than complete theoretical distributions. The success of these approaches unveils the non-uniform nature of the probability space.

2.1. Evidence of non-ergodicity in language models

Ergodicity in dynamical systems implies that a system, if left to itself for long enough, will pass close to almost all the dynamical states consistent with energy conservation. Though this is a very simplistic view to define ergodicity. A dynamical system may have a hierarchy of properties, each of which implies the one before it[39]. Ergodicity is only the first. Central to understanding ergodicity is the notion of symmetry in temporal evolution[40][41]. The invariance of statistical properties under time translation in ergodic systems reveals time symmetry; the system's behavior remains consistent whether observed now, at a later time, or in a forward or backward temporal direction[42][43]. This temporal symmetry guarantees that time averages are equivalent to ensemble averages, a fundamental principle of statistical mechanics[44]. Non-ergodic systems break this symmetry, which fundamentally influences the system's future possibilities and creates distinct temporal phases that are impossible to average[45].

Consider how language models generate text: each token prediction depends not just on direct context, but on the entire sequence of previous tokens. Unlike classical ergodic systems, where future states are independent of the path taken to reach the current state, language models exhibit strong path dependence. A word appearing early in a sequence can fundamentally alter the probability distribution of all subsequent tokens. This creates an intrinsic asymmetry in time that violates the basic premise of ergodicity. For example, when a language model builds a story, the context and characters selected at the outset limit all potential scenarios. Past choices introduce semantic and logical constraints for coherence, preventing the model from exploring all possible story states. This is similar to how biological systems expand within constrained spaces, where every possibility during development bounds and shapes what might happen in the future. Language models show this non-ergodic behavior by using their attention mechanisms and analysis of context. The cumulative context window shapes the model's state space, creating what we might call 'semantic valleys' that guide and constrain predictions about future tokens. Although time and ensemble averages converge in ergodic systems, language models have persistent memory effects that make certain semantic paths more probable based on the past evolution.

Meister & Cotterell[11] observation that models learn "only a subset of the tendencies" rather than complete theoretical distributions provides clear evidence for non-ergodicity in language models[14][32][39][46]. This would imply that models follow a constrained exploration, thus not operating in a fully ergodic space where all states are equally accessible[47]. This aligns with Kauffman's theory of constrained possibility spaces[18]. Language models, as complex biological systems, show a preference for empirically observed patterns over theoretical possibilities[32]. These models also capture path-dependent patterns emerging from actual language use. The fact that the probability space is shaped by training history rather than theoretical distributions creates a fundamental asymmetry in how models explore their possibility space[17].

3. Complex dynamics and emergence in language models

The evolution of probability distributions across tokens shows patterns that go beyond simple statistical dependencies. For example, classical statistical measures like perplexity fail to capture emergent capabilities in language models[2]. This limitation is not new in complex systems, where reductionist statistical methods fail to capture emergent behaviors[48].

3.1. Limitation of classical statistics

In systems exhibiting non-linear behaviors resulting from the interaction of multiple parts or sub-systems, basic aggregation of probabilities fails to describe coherent, long-range dependencies. In LLMs, complex interactions between context layers create capabilities not predictable from individual components[3]. These capabilities often appear swiftly at certain scales[2], suggesting phase transitions in model behavior[32][46][39]. In language models, probability distributions are likely to evolve through paths that preserve coherence throughout long sequences, suggesting they operate as complex adaptive systems (CAS). In CAS, non-linear behaviors take place from multiple interacting components showing a high sensitivity to initial conditions. These systems also present self-organizing properties emerging at multiple scales.

3.2. Complex system dynamics and emergent behaviors in language models

The emergence of capabilities in language models suggests some characteristics common in complex adaptive systems, such as hierarchical organization and phase transitions[39]. Token-level interactions give rise to higher-order semantic structures where multiple scales of organization emerge simultaneously[49]. These hierarchies resist reductionist analysis. LLMs demonstrate capabilities that emerge at certain model scales[2]. These emergences imply phase transitions, revealing the presence of critical phenomena in model behavior.

Recent research demonstrates that context significantly influences model behavior due to the creation of context-dependent representations through sequential processing[3]. Dynamic memory effects influence long-range dependencies[50][51], and context modifications show non-linear effects on model output[4]. These mechanisms suggest an adaptive behavior where models seem to adapt to changing contexts. Furthermore, long-range coherence emerges from local interactions[3] and self-organization appears at multiple scales[2].

4. Complex adaptive systems and biological evolution

Complex adaptive systems, or CAS, are defined by some fundamental mathematical properties that discern them from simple dynamical systems[52][53]. CAS create and apply internal models to predict the future, taking current actions according to expected outcomes. This characteristic differentiates CAS from other types of complex systems, as well as making the emergent behavior of CAS more difficult to understand[54]. When Haken[52] coined the term “synergetics,” he gave a very simple definition, referring to self-organizing systems (a property of CAS) as those characterized by the fact that the system finds its organization or function on its own, without direct external guidance[55].

Analogously, language models develop internal representations during pre-training, capturing statistical patterns of language, semantic relationships, contextual dependencies, and domain knowledge. These internal models are encoded in the weights and attention patterns of the neural network[35]. Internal models are able to generate next tokens based on predicted probability distributions using attention mechanisms to “look back” at context and then “predict forward.” The model's decisions about token selection are based on these predictions that can also be changed based on the evolving context. As a result, language models' capabilities "emerge" from interactions between different layers, attention heads, and learned patterns. During their training, language models adjust weights based on training text data, and during inference, they adapt their internal model to the specific context of the current conversation or task.

An important practical limitation of CAS is that they don't have a single governing equation, or rule, that controls the system[53]. Thus, a direct approach to analyze these systems is by evaluating their different properties, such as non-linearity, emergence, self-organization, and phase transitions[56]. Non-linearity implies that the system's behavior cannot be derived from linear relations[56]. The following sections will elaborate on the meaning of emergence, self-organization, and phase transitions.

4.1. Emergence in complex systems

Emergence appears when complex behaviors arise from simple rules and interactions in a system[48]. The CAS theory[57][53][58] together with synergetics[52] provide a comprehensible theoretical framework that can be used to study emergence. Mathematically, emergence can be formalized through the interaction between fast and slow variables in a system. Slow variables are the high-level patterns that emerge and guide the system, meanwhile fast variables are the detailed, moment-to-moment changes in the system[59]. We can consider these variables as the macroscopic parameters[58]. In these systems, the link between order parameters and components is complex because multiple components (fast variables) influence, and sometimes define, the order parameters. This is known as the slaving principle, which results in the notion of circular causality. The limited order parameters govern the behavior of the individual components, whereas the components influence the behavior of the order parameters[52]. Haken[60] defined the slaving concept, which links both rapid and slow variables, as follows:

$$\frac{dq}{dt} \equiv \dot{q} \; = N(q,\; \nabla, \alpha) + F(t) \tag{1}$$

where q is the state vector (microscopic level variables), N is a nonlinear vector function, ∇ is a grading operator acting on q, α represents control parameters, and $F(t)$ denotes fluctuating forces that characterize the external or internal noise affecting the system. The transition from Equation 2 to a simpler parametric equation describing the system's collective behavior is not evident. The complete mathematical derivation, including all necessary assumptions and detailed proofs, can be found in the literature[61][62][63][64]. In short, we need to find a time-independent solution q^0 for a specific set of control parameters. Then, when the system operates near an instability point, small perturbations around this solution can be analyzed. This allows us to sort the system's behavior into different time scales, identifying fast-decaying stable modes and slowly-evolving unstable modes that become critical near the instability point. In a language model, slow variables would be equivalent to the overall flow of a story in language generation, and fast variables could be individual word choices.

4.2. Self-organization and phase transitions

In statistical mechanics, the underlying assumption behind the theory of self-organized criticality (SOC) is that a complex system will naturally organize itself into a state on the edge of two different regimes, without intervention from outside the system[65][66]. The mathematical formalization of self-organizing systems can be expressed through the concept of pattern formation and symmetry breaking[67]. Systems exhibit spontaneous pattern generation governed by equations of this type:

$$\frac{\partial u}{\partial t} = D\nabla^2 u + f(u, \alpha) \tag{2}$$

where u represents the pattern-forming field, D is a diffusion coefficient, and α is a control parameter. When α reaches critical values, the system undergoes spontaneous symmetry breaking, leading to pattern formation. Function u could represent, for example, temperature variations in thermal convection, or population density in ecological systems. In language models, u could represent the distribution of attention weights, the activation patterns across layers, or the probability distributions over tokens. Equation 2 is divided into two terms: $D\nabla^2 u$ calculates how the field changes over time from diffusion or spatial spreading, and $f(u, \alpha)$ represents the local dynamics.

Phase transitions are another fundamental property of complex adaptive systems, marked by abrupt changes in system behavior at critical points[68]. The universality principle categorizes different physical systems based on their behavior near critical points, leading to the emergence of universal scaling laws, also known as power laws[69]. Physical quantities follow power laws as systems approach critical points. At these phase transitions, key parameters like correlation length and susceptibility show divergent behavior, characterizing the critical phenomena. The correlation length ξ near a critical point, T_C , represents the scale at which a system's general properties begin to diverge from its main properties. It can be defined as $\xi \sim |T - T_C|^{-\nu}$, where ν is the critical exponent governing the divergence of the correlation length. The characteristic length scale ξ diverges as the system T approaches a critical point T_C with a negative exponent $-\nu$ by following a power-law behavior.

In language models, self-organization arises through the emergence of coherent text structures from local token interactions, whereas phase transitions are characterized by sudden improvements in model capabilities at certain scales[70][71]. The correlation between successive tokens follows power-law scaling near critical points, suggesting similar underlying mechanisms to phase transitions occurring in many natural phenomena.

4.3. Non-ergodicity in biological systems

The ergodic hypothesis articulates the notion that a point within a moving system, whether it be a dynamical system or a stochastic process, will eventually go through every part of the space in which it works, in a way that is both uniform and random[43]. This suggests that we can infer the overall behavior of the system from the path taken by a representative point. Classical statistical mechanics relies on the ergodic hypothesis, which states that time averages equal ensemble averages[39]. We can define an ergodic system as one in which, for any property A, the time average and ensemble average are equivalent: $\overline{A} = \langle A \rangle$. In ergodic systems, events occur quickly relative to an observation time $\tau_{int} \ll t_{observed}$. When the system may be evolving at a very slow rate too for an observer ($\tau_{int} \gg t_{observed}$), the system enters a non-ergodic state. The hypothesis that, given enough time, a system will explore its entire phase space implies that a system will eventually explore all accessible states with equal probability. Ergodic breakdown can be probed by either measuring the evolution of τ_{int} for some properties at fixed $t_{observed}$, or by changing $t_{observed}$ for fixed τ_{int}[72].

Biological systems challenge this assumption via two primary mechanisms: historical contingency, since the system's current state depends critically on its past trajectory and not only on the current configuration[73], and through adaptive dynamics, where the whole space of possible states evolves as the system advances[18]. These mechanisms can be formalized with the following equations:

$$P\left(st+1|st\right) \neq P\left(st+1\middle|s^{'}t\right) \tag{3}$$

$$\Omega(t+1) \neq \Omega(t) \tag{4}$$

Equation 3 shows that, even when st and $s^{'}t$ have the same energy, path-dependent transition probabilities P differ due to historical unfolding[74]. Equation 4 shows the adaptive dynamic nature of the phase space of accessible states Ω. According to these mechanisms, biological systems cannot be understood through the statistical ensemble averages[75].

4.4. The adjacent possible theory (TAP)

Theoretical biologist Stuart Kauffman worked to figure out fundamental principles that govern a specific category of non-equilibrium systems, particularly those involving coevolutionary self-constructing communities of autonomous agents[5]. The adjacent possible theory, or TAP, appeared as an important advance for understanding how biological and other complex systems explore and expand their possibility spaces. In his book "Investigations," Kauffman presents this concept by initially considering an important question: How do biological systems perpetually generate novelty in an apparently limitless way? The solution lies in understanding how each current state of a system defines a collection of possible subsequent states—a concept he refers to as the adjacent possible. The adjacent possible suggests not every conceivable state, but specifically those states that exist just one step away from the present reality, unveiling the potential transformations that can arise from the existing organization[76]. However, it is important to note that, in contrast to phase spaces in physics, each expression of an adjacent possible state generates new additional adjacent possibles. In Kauffman's words, "The adjacent possible consists of all those molecular species that are not members of the actual but are one reaction step away from the actual"[5].

TAP provides a theoretical framework for understanding how systems can be both constrained by their current state and perpetually creative. It suggests that evolution, rather than exploring a fixed space of possibilities, indeed expands the very space of what is possible. This expansion follows what Kauffman defines as "the laws of the construction of the possibilities of the biosphere"[5]. TAP also provides a mathematical framework for understanding non-ergodic evolution in biological systems[18].

According to TAP, complex systems evolution could be described with the following equation[32]:

$$M_{t+1} = M_t + \sum_{i=1}^{M_t} \alpha^i \binom{M_t}{i} \tag{5}$$

where M_t represents the number of elements in the phase space at a given time t (a molecule, a species in an ecosystem, an innovation in the market, etc. The constant α, $0 \leq \alpha \leq 1$, is a constraint parameter that limits which combinations are allowed. When $\alpha = 1$ there is an evolution of the total possible throughout time. Exponent i represents an index for summation over possible combinations. Constant α increased to the power of i operates as a limiting factor for larger combinations. The α parameter in the TAP equation constraints the combinations that are likely to occur, establishing a balance between deterministic factors (combinations must be feasible given the current M_t) and stochastic exploration (the actual combinations that occur will depend on α). The binomial coefficient (M_t choose i) provides the potential combinations of i items selected from M_t items. Equation 5 describes the emergence of new possibilities through the combination of existing elements, while setting constraints on the accessibility of greater combinations. This conceptual framework shows how evolution emerges through accessible adjacent states rather than random leaps, representing the dynamic nature of phase space: each new element M_{t+1} creates new combinatorial possibilities and the phase space dimension grows as new combinations become accessible.

Latest Kaufmann's work rewrites Equation 5 in a different manner. Instead of considering α as a constant increased with exponent i, this constraint does not depend on a single constant value but on a sequence of constraint constants[77][78][31].

$$M_{t+1} = M_t + \sum_{i=1}^{M_t} \alpha_i \binom{M_t}{i} \tag{6}$$

This reformulation represents a more realistic description of biological and other adaptive complex systems where the possibilities space is constrained by several factors. For example, in the evolution of metabolic networks in cells, multiple constraints can operate simultaneously. Chemical constraints will regulate which reactions are thermodynamically possible, while enzymatic constraints will limit catalytic chemical reactions. Additionally, the metabolites available will complete the resources constraints. Each new metabolic innovation M_{t+1} is constrained not by a single factor but by the interaction of these multiple constraints. A new metabolic pathway might be chemically possible (high α for chemical constraints) but limited by the availability of specific enzymes (low α for enzymatic constraints).

Architectural, training, and contextual factors constrain the space of possible token predictions in large language models. This parallels the way in which the aggregation of multiple constraints expands the metabolic possibility space.

5. Application of TAP framework to language models

The conceptual framework introduced by the TAP equation could be applicable to understanding how language models could navigate their possibility spaces via restricted combinations of existing elements rather than through a random search of all potential possibilities.

5.1. Mathematical framework

We need to define the necessary mathematical structures for mapping TAP to language models. Let (Ω, F, P) be the probability space associated with language model token predictions, where Ω is the sample space of all possible token sequences, F is the σ-algebra of measurable events[79], and P is the probability measure generated by the model.

5.1.1. Model's state space

While biological systems operate in continuous state spaces, language models work with discrete token sets. Let M_t represent the state of the system at time t. For a language model, we can define M_t as a tuple (V_t, S_t, P_t) where $V_t \subseteq V$ is the active vocabulary subset at time t, S_t is the semantic state space at time t, and $P_t : V_t \longrightarrow [0,1]$ is the probability distribution over tokens. For a mapping between continuous semantic representations and discrete tokens, we define the discretization operator $D_t : S_t \longrightarrow V_t$. Formally, we can define S_t as a manifold in a high-dimensional space:

$$S_t = \{s \in \mathbb{R}^n | \exists \, \varphi : V_t \longrightarrow \mathbb{R}^n \; and \; D_t \circ \varphi\} \tag{7}$$

where φ is smooth and locally invertible. The dimensionality of S_t, $dim(S_t)$ is the number of independent semantic features actively involved in token prediction at time t.

In language models, while the lexical space V is constrained by a fixed vocabulary or lexicon, the semantic space S reveals a complex system with hierarchical organization[80]. Just as a book represents a hierarchy from words to complete narratives, language models process and generate language across multiple hierarchical levels: starting from individual tokens as elementary units to phrases, clauses, sentences, paragraphs, and even broader narrative structures. These models create this organization through attention mechanisms and contextual relationships[81], where each level emerges from combinations of lower-level elements. The possibilities for these combinations expand as we move up the hierarchy. This is analogous to the slaving principle defined by Haken[59], where fast variables (lower-level elements) generate slow variables (high-order elements).

doi.org/10.32388/00JQN6

5.1.2. Computational resources space

We can define computational resource utilization as the vector-valued function $C_t = (Mem_t, A_t, H_t)$, where $Mem_t \in \mathbb{R}^+$ is the model memory use, $A_t \in \mathbb{R}^+$ is the attention computation cost, and $H_t \in \mathbb{R}^+$ is the hidden state computation cost at time t. The total computational cost can be represented as:

$$R(C_t) = min\left(1,\ \frac{C_{\max} - C_t}{C_{\max}}\right) \tag{8}$$

where $\|C_t\| = w_1 Mem_t + w_2 A_t + w_3 H_t$ is a weighted norm, $C_{\max}$ is the maximum computational capacity, and w are weight coefficients for different resource types. The resource bound function $R(C_t)$ connects to model architecture as follows:

$$R(C_t) = min\left(1,\ \prod_{i=1}^{n} r_i\,(C_t)\right) \tag{9}$$

where $r_i\,(C_t)$ is the individual resource constraints from memory capacity, attention computation, context window size, and hidden state dimension.

5.1.3. Mapping TAP equation to language models

Lemma 1. Let V be the vocabulary space and S be the semantic space of a language model. There exists a measurable mapping

$$\varphi : V \times S \to \Omega \tag{10}$$

that satisfies the following properties:

- For any state space $M_t \in V \times S$, the mapping preserves the combinatorial structure of Equation 6 from TAP

  $$M_{t+1} = (V_{t+1}, S_{t+1}, P_{t+1}) \tag{11}$$

  where

  $S_{t+1} = \ \varphi(S_t) \cup \left\{\varphi\left(\sum_i \alpha_i \begin{pmatrix} M_t \\ i \end{pmatrix}\right)\right\}$

  $V_{t+1} = D_t\,(S_{t+1})$

  and $P_{t+1} : V_{t+1} \longrightarrow [0, 1]$
- For any $x \in V \times S$, the mapping is bounded by computational resources:

  $$\|\varphi(x)\|_2 \leq R\,(C_t) \tag{12}$$

- The mapping preserves the dimensionality constraints:

$$\dim(Im(\varphi)) \leq \dim(V) \times dim\left(S_t\right) \tag{13}$$

Proof. Given the measurement space $(V \times S, B, \mu)$ where B is the Borel $\sigma-$algebra, and μ is the product measure[79]. We define φ through the composition $\varphi = \pi \circ \psi$, where ψ is the attention mechanism, and π is the projection onto the probability simplex. The mapping φ preserves TAP equation structure:

$$\varphi\left(M_t + \Delta M\right) = \varphi\left(M_t\right) + \nabla\varphi\left(M_t\right)\Delta M + \mathrm{O}\left(\|\Delta M\|^2\right) \tag{14}$$

Equation 14 illustrates that when we make small changes to the model's state, the resulting changes in the mapped space are well-behaved and predictable - they consist mainly of a linear component plus some small higher-order corrections. This is central to showing that the mapping is compatible with how the TAP equation describes system evolution. The equation essentially establishes that φ is differentiable and provides a Taylor expansion around M_t, which is necessary for proving the mapping preserves the mathematical structure needed for the TAP framework.

5.1.4. Resource limits

Lemma 2. For the mapping $\varphi: V \times S \rightarrow \Omega$ defined in Lemma 1, there exists a positive constant K such that:

$$\|\varphi\|_2 \leq K \bullet R\left(C_t\right) \forall x \in V \times S \tag{15}$$

The proof of this lemma can be developed in three basic steps, each dependent on the previous one in order to set a fitting bound. First, we are going to define the combinatorial framework of token prediction in large language models and its relation to TAP theory. Second, we will define an isomorphism between attention processes and the combinatorial space of TAP. Finally, we will verify resource boundedness through analyzing model capacity constraints.

First step. Next token prediction in LLMs follows a combinatorial structure analogous to TAP. The probability of the next token given a context can be represented as:

$$P\left(token_{t+1} | context\right) = \sum_i w_i \bullet g\left(tokens_t\right) \tag{16}$$

where g denotes the attention mechanism operations and w_i are learned weights. This structure directly corresponds to the combinatorial summation in TAP.

Second step. The attention mechanism provides a natural isomorphism to TAP's combinatorial space. For any query Q, keys K, and values V:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \cong \sum_i \alpha_i \binom{M_t}{i} \tag{17}$$

This isomorphism is valid because the softmax function constrains outputs to the interval $[0, 1]$, similarly to Kauffman’s TAP’s α coefficient in equations 5 and 6. In addition, the distribution of attention patterns reflects TAP equation combinatorial selection, and the scaling factor $1/\sqrt{d}$ guarantees numerical stability by providing a natural limit.

Third step. We can define resource boundedness through model capacity constraints $\dim(Attention) \leq C_{\max}$. This evidences the finite dimensionality of attention head outputs and the bounded nature of the weighted sum across heads by considering the resource constraint $R\left(C_t\right)$. Hence, integrating these results:

$$\begin{gathered} \|\varphi\|_2 = \|Attention\left(Q(x), K(x), V(x)\right)\|_2 \leq \\ \|V\|_2 \bullet \left\|softmax\left(\frac{QK^T}{\sqrt{d}}\right)\right\|_2 \leq K \bullet R\left(C_t\right) \end{gathered} \tag{18}$$

where $K = max\left(||V||^2\right) \cdot \sqrt{(\dim(V))}$.

Corollary. The resource boundedness of φ implies that the semantic space S_t grows at a rate constrained by available computational resources:

$$\frac{\partial dim\left(S_t\right)}{\partial t} \leq h\left(R\left(C_t\right)\right) \tag{19}$$

where h is a monotonic function of the resource bound. Lemma 2 and this corollary establish the mathematical basis for describing how computational resources limit the expansion of the semantic space in language models, connecting theoretical TAP dynamics in complex biological systems with large language models constraints.

5.1.5. Semantic space evolution

To model the semantic space evolution, we have to extend Kauffman's idea of expanding possibility spaces by explicitly incorporating computational limitations. The evolution of the semantic space can be modelled with the following equation:

$$\frac{d}{dt}\dim(S_t) = \sum_{l=1}^{L} \nabla g_l \frac{\partial C}{\partial t} - \lambda(t)\dim(S_t) \tag{20}$$

where $\dim(S_t)$ represents the number of dimensions the semantic space has at time t, and $\lambda(t)$ is a decay term ensuring computational feasibility. The term ∇g_l captures how the hierarchical functions influence dimensionality at each level l.

5.2. Constraints in language models

Equation 6 provides a framework that can be naturally mapped to token space growth. We identify three key types of constraints that influence language model behavior: architectural constraints[82][35], training data constraints[7], and contextual constraints[3]. These constraints interact to shape model capabilities and performance. We are going to describe these constraints in the following sections.

5.2.1. Architectural constraints

Some limitations set by the model's architecture are the vocabulary size, which limits possible tokens[82], and the context window length, which restricts the amount of past information that can influence predictions[3]. The model's attention mechanism design constrains through both computational and architectural limitations. The mechanism's quadratic complexity with sequence length creates memory and speed constraints, while the structure of attention heads limits parallel relationship tracking through the trade-off between head count and dimensional capacity[83]. The model's information flow is also limited by attention patterns and inter-token path length, which control the ability to capture relationships[35].

The multi-head architecture enables the model to simultaneously analyze many text features, yet the fundamental configuration of these attention patterns fixes post-training[35]. While increasing the number of attention heads enhances the model's ability to capture diverse relationships in the text, it also reduces the depth of each head's representation of these interactions. This eventually results in a considerable reduction in the model's capabilities. The distribution of restricted computing resources between the extent of attention coverage and the depth of connection representation causes this limitation[84].

5.2.2. Training data constraints

Training data constraints can include statistical patterns in language[85][7], domain-specific knowledge acquisition[86], implicit learning of grammar and syntax[87], and token co-occurrence patterns[82]. In neural networks, the emergent linguistic hierarchical structure represents a basic constraint, showing implicit patterns and rules learned during training[88].

The learned connections between tokens and their contexts, and how often they appear together in different contexts, also limit the model by shaping the probability space for predicting the next token, becoming an important constraint to consider[7].

5.2.3. Contextual constraints

Contextual constraints complete how language models predict the next token in a sequence. These constraints shape token prediction through three primary mechanisms: First, sequential dependencies operate through attention mechanisms, where each token's prediction is influenced by all previous tokens in the sequence[35]. This enables the model to maintain coherence over long sequences by incorporating the full context of prior outputs[3]. Second, models develop internal representations that track semantic relationships throughout the generated text[2]. These representations help maintain topical coherence by preserving key semantic information across the generation process, preventing topic drift and ensuring contextual relevance. Third, style consistency emerges from the model's ability to recognize and maintain attributes like tone and formality. This style coherence is achieved through pattern recognition learned during training[82], and reinforced through contextual processing. The interaction of these mechanisms creates a dynamic constraint system that guides text generation while preserving semantic and style consistency.

5.2.4. Constraints interaction mechanism

Equation 6 for the TAP framework introduces the constraint factor α_i, which is a combination of several constraint factors. Therefore, to translate the TAP equation to language models, we need to define how these constraints interact. To clarify the nature of these interactions, we could look at how different complex biological systems' mechanisms work. For example, in biological metabolic networks, the Michaelis and Menten equation for enzyme kinetics follows multiplicative interactions to integrate different rate constants[89]. The apparent equilibrium constant in the Michaelis and Menten equation is derived from the product of the ratios of the forward and reverse rate constants for each reaction step[90].

In gene regulation, there is evidence for both additive and multiplicative interactions between transcription factors that regulate the transcription rate of a set of target genes[91]. Some of these factors can follow multiplicative effects, meanwhile others can be combined through additive effects. In cell signaling, there is further evidence of both multiplicative and additive signal integration[92]. For example, in calcium signaling, cells use different types of Ca^{2+} influx channels to contribute to the cytoplasmic calcium increase. These inputs present different activation mechanisms as voltage-operated (VOOC), receptor-operated (ROOC), mechanically activated (MA), and stock-operated (SOOC)[93]. The combination of these mechanisms implies an additive process where $\left[Ca^{2+}\right]_{influx}$ is the sum of the different channels[94]. An example of a multiplicative effect in cell signaling can be found in the biological responses associated with mitogen-activated protein kinase (MAPK) signaling[95].

When sequential dependent processes exist, multiplicative interactions prevail. The already mentioned Michaelis-Menten equation is a classic example of an enzyme cascade in which each step is dependent on the completion of the previous phase. As a general rule, in equilibrium-based systems, the multiplicative mechanism prevails. Additive interactions prevail in parallel and independent processes where multiple paths can achieve the same outcome.

We could conclude by saying that, in biological systems with alternative pathways or where multiple processes share resources through compensation mechanisms, the additive process is prevalent. Conversely, in systems that incorporate redundancy mechanisms, a multiplicative mechanism prevails, enhancing the system's robustness.

We could apply the TAP equation to language models in three ways: either by linking the sequence of constraint factors $\alpha_i$additively, multiplicatively, or simultaneously by both mechanisms. In language models, architectural constraints appear to be multiplicative because all components are necessary for the model to operate efficiently; if any architectural component (vocabulary access, attention mechanism, or context processing) fails, the model fails. In contrast, training data constraints might follow additive patterns since multiple different training examples can lead to similar model behaviors. Context constraints could theoretically exhibit both behaviors: they can multiply when the context demands strict requirements like logical flow, reference resolution, or grammatical consistency. However, when the context offers multiple valid paths, such as different synonyms that achieve the same meaning, alternative phrasings that maintain style, or multiple valid continuations of a story, these elements combine additively, representing parallel valid options.

Based on these findings, we propose focusing on the multiplicative effect when applying the TAP equation to language models for several reasons:

- It better captures the critical nature of architectural constraints,
- aligns with how biological systems handle essential component interactions, and
- provides a more conservative estimate of possibility space growth.

5.2.5. Constraints and non-ergodicity in language models

We have considered three natural constraints influencing language models' state space expansion. Let $T:\ M_t \to C_t$ be the mapping from state space to resource space. Then we have:

$$T\left(Mem_t\right) = \left(\|Mem_t\|_{mem}, \|A_t\|_{comp}, \|H_t\|_{state}\right) \tag{21}$$

where Mem_t is the model's memory use, A_t is the attention computation cost, and H_t is the hidden state computation cost at time t. Each component of T induces our natural constraints:

$$\beta_i = sup\left\{x : \|Mem_t\|_{model\ memory} \leq C_{\max}\right\} \tag{22}$$

$$\gamma_i\ = sup\left\{x : \|A_t\|_{attention} \leq C_{\max}\right\} \tag{23}$$

$$\delta_i\ = sup\left\{x : \|H_t\|_{hidden\ state} \leq C_{\max}\right\} \tag{24}$$

where β_i are the architectural constraints, γ_{i} are the training data constraints, and δ_{i} are the contextual constraints. These constraints converge into an overall constraint function $\alpha(i,t)$, which is dependent on time due to its dynamic nature.

The ergodic hypothesis postulates that the system spends equal times in equal volumes of its fixed phase space[14]. Next token prediction in language models operates at two fundamental levels: the lexical level and the semantic level. In language models, the lexical formatives are selected in a well-defined way from a fixed universal vocabulary set[96]. The current mathematical framework demonstrates the non-ergodic nature of language models through three primary mechanisms:

- Path-dependent resource use. Given two states s_t, s'_t with equal computational cost $\|C_t\|$, their future resource utilization differs:

$$P\left(C_{t+1}|s_t, R(C_t)\right) \neq P\left(C_{t+1}|{s'}_t, R(C_t)\right) \tag{25}$$

  This path dependence introduces the architectural constraints β_i through:

$$\beta_i = sup\left\{x :\ P\left(C_{t+1}|s_t, x\right)\right\} \tag{26}$$

This dependence reflects how attention mechanisms allocate computational resources based on context history, creating inherent asymmetries in resource utilization.

- Training-induced state space restrictions. The training process creates fundamental asymmetries in how the model explores its state space $\Omega_{t+1} \neq \Omega_t \cup \{new\ states\}$, leading to training constraints $\gamma_i = sup\{x:\ x \in \Omega_t \cup \Omega_{t+1}\}$. These constraints emerge from the model's learned patterns and knowledge representations, affecting how it navigates its possibility space.

- Context-dependent transitions. The semantic component governs the model's word interpretation according to a context, creating a path-dependent behavior that can be defined as $P\left(s_{t+1}|s_t\right) \neq P\left(s_{t+1}|{s^{'}}_{t}\right)$. This dependence implies emergent contextual constraints that can be defined as $\delta_i = sup\{x:\ P\left(s_{t+1}|s_t, x\right)\}$. For the system state space $M_t = \left(V_t, S_t, P_t\right)$, these mechanisms can be seen in output probability transition \(P\left(M_{t + 1}|M_{t} \right) \neq P\left{ M_{t + 1}|{M^{'}}_{t} \right)\) revealing a path dependence even when M_t and M'_t have identical computational requirements. This path dependence can be observed in:
  a. The model's attention weights evolve as well based on both current and historical context: $W_t = f(W_{t-1}, X_t)$ where X_t is the current input. This evolution creates memory-like effects in the model's processing.
  b. Even when the aggregate information content is identical $\sum_{i=1}^{n-1} t_i = \sum_{i=1}^{n-1} t_i^{'}$, different sequence orderings produce different probability distributions $P(t_n|t_1, \ldots, t_{n-1}) \neq P(t_n|t'_1, \ldots, t'_{n-1})$ because order information presentation influences model behavior.
  c. The model's computational resource use depends on the specific path taken $R(C_t|h_t)/R(C_t|h'_t)$ where h_t is the processing history. This creates a fundamental asymmetry in resource allocation based on the specific sequence of previous states.

The semantic manifold S_t evolves non-uniformly as:

$$S_t = \{s \in \mathbb{R}^n | \exists \varphi : V_t \rightarrow \mathbb{R}^n\ such\ that\ s = \varphi(v)\ for\ v \in V_t\} \tag{27}$$

where φ is a smooth mapping function taking tokens to semantic vectors. We can define $\varphi(v)$ as follows:

$$\varphi(v) = D_t^{-1}\left(Attention\left(W_Q Q_v, W_K K_{context}, W_V V_{context}\right)\right) \tag{28}$$

where D_t^{-1} is the inverse mapping from token to semantic space, Q_v is the query vector for token v, $K_{context}$ is the key matrix for the context, $V_{context}$ is the value matrix for the context, and W_Q, W_K,

W_V are learned weight matrices. This mapping is non-uniform since attention weights depend on the entire context history, the same token can map to different semantic vectors depending on context, and the dimensionality of S_t can change as context accumulates

These non-ergodic qualities automatically give rise to the three types of constraints (architectural, training, and contextual) that characterize the model's behavior, building a direct connection between the system's non-ergodicity and its operational constraints.

5.3. A TAP equation for language models

We propose a version of the TAP equation for modeling the expansion of language models, taking into account the specific constraints and dynamic nature. To arrive at this equation, we have to define the phase space (equivalent to M_t in Equation 6) and the integrated constraint function, equivalent to α_i in Equation 6.

5.3.1. Resource-bounded phase space

Let A_t represent the accessible state space at time t. We first establish the fundamental resource bound:

Lemma 3. For any language model with maximum computational capacity $C_{\max}$, there exists a monotonic function f such that $\sup\left(A_t\right) \leq f\left(C_{\max}\right)$ where $f(x) = \kappa \cdot x \cdot log(x)$, and κ is a model architecture-dependent constant. The bound is defined by the following factors:

- memory constraints $\mathcal{O}\left(d_{model} \cdot n_{layers}\right)$,
- attention computation bounds $\mathcal{O}\left(sequence^2\right)$, and
- vocabulary access limitations $\mathcal{O}\left(|V| \bullet d_{model}\right)$.

5.3.2. Constraint integration

We define the integrated constraint function as

$$\alpha(i,t) = min(\beta_i, \gamma_i \delta_i, R(C_t)) \tag{29}$$

where β_i means architectural constraints, γ_i are training data constraints, δ_i are contextual constraints, and (C_t) is the resource bound function. The multiplicative relation is justified as all components must operate effectively for the model's complete functionality. The integrated constraint function $\alpha(i,t)$ in our TAP equation can be defined as:

$$\alpha(i,t)\prod_{j=1}^{m}c_j(i,t) \tag{30}$$

where $c_j(i,t)$ are the individual constraints, and m is the number of active constraints.

5.3.3. Proposed resource-bounded TAP equation

In Equation 6, α_i represents a fixed constraint on combinatorial possibilities according to initial resource availability. Lower values of α at $t=0$ indicate a system with limited starting resources, hence constraining its future possibility of growth relative to systems with higher α values. Large language models have two types of fixed initial constraints: a fixed universal vocabulary set $|V_t|$, and fixed computational resources driven by the model architecture (memory capacity, attention heads, and context window size). The third type, contextual constraints, differs fundamentally as they begin minimal and evolve dynamically as context accumulates during operation. The semantic space evolution thus lacks a fixed initial constraint at $t=0$, reflecting the dynamic nature of constraints in language models. This dependence on initial conditions, particularly for fixed constraints, is a common property of complex adaptive systems, CAS[68][58][97].

To arrive at our equation, we must follow several important steps. Extending TAP's central idea in Equation 6, we integrate the computational resources $R\left(C_t\right)$:

$$M_{t+1}=M_t+R\left(C_t\right)\sum_{i=1}^{M_t}\alpha_i\binom{M_t}{i} \tag{31}$$

We split α_i into component constraints as detailed in (29):

$$M_{t+1}=M_t+R\left(C_t\right)\sum_{i=1}^{M_t}\left(\beta_i,\gamma_i\delta_i\right)\binom{M_t}{i} \tag{32}$$

We need to add hierarchical structure using a special function g_l while incorporating $R\left(C_t\right)$ into the hierarchical function's bounds: $\|g_l(x)\|_2\leq K\cdot R\left(C_t\right)$. This function maps from $P(V)$, which is the probability space over the vocabulary V, to an n-dimensional real space $P(V)\rightarrow\mathbb{R}^n$

$$M_{t+1}=M_t+\sum_{l=1}^{L}g_l\sum_{i=1}^{M_t}\left(\left(\beta_i,\gamma_i\delta_i\right)\binom{M_t}{i}\right) \tag{33}$$

At time $t=0$ $M_t=|V_t|$, showing that the initial state space is constrained by the fixed vocabulary size. This initial condition reflects the starting point where only lexical combinations are possible, before the semantic space begins its dynamic evolution through hierarchical interactions and constraint effects. Finally, we replace the term $(\beta_i, \gamma_i \delta_i)$ by the constrain function $\alpha(i,t)$. This leads to our final resource-bounded TAP equation for language models:

$$A_{t+1} = A_t + \sum_{l=1}^{L} g_l \sum_{i=1}^{|V_t|} \left(\alpha(i,t) \binom{|V_t|}{i} \right) \tag{34}$$

This equation describes how language models explore their possibility space while subject to constraints. A_{t+1} represents the accessible state space at the next time step, and A_t is the current accessible state space. The first sum $\sum_{l=1}^{L} g_l$ captures the hierarchical levels of language processing (from tokens to phrases to broader structures). The second sum $\sum_{i=1}^{|V_t|}(\bullet)$ represents all possible combinations within the vocabulary size. $\alpha(i,t)$ combines all constraints (architectural, training, and contextual) at time t, and the binomial coefficient ($|V_t|$ choose i) represents possible combinations of tokens.

The hierarchical function g_l transforms these combinations into the model's semantic space bounded by computational resources through the condition $\|g_l(x)\|_2 \leq K \cdot R(C_t)$. Constant K is a model architecture-dependent constant that scales the relationship between the hierarchical function and computational resources. It sets an upper bound on how much the hierarchical transformations can expand given the available resources. We have defined this constant in Equation 18 as $K = max(\|V\|^2) \cdot \sqrt{dim(V)}$, where $max(\|V\|^2)$ describes the maximum norm of vocabulary embeddings, and $\sqrt{dim(V)}$ is the dimensionality of the vocabulary space.

Equation 34 satisfies the following conditions:

a. Conservation

$$\partial A_t / \partial t \leq g\,(C_{\max} - \|C_t\|) \tag{35}$$

Let

$$A_{t+1} - A_t = g_l \textstyle\sum_{i=1}^{|V_t|} \left(\alpha(i,t) \binom{|V_t|}{i} \right) \tag{36}$$

Then:

$$\frac{\partial A_t}{\partial t} = \lim_{h \longrightarrow 0} \frac{A_{t+h} - A_t}{h} = \frac{\sum_{l=1}^{L} g_l \sum_{i=1}^{|V_t|} \left(\alpha(i,t) \binom{|V_t|}{i} \right)}{h} \tag{37}$$

where h represents an infinitesimal time step in the limit calculation. Considering the bounds introduced before, we have:

$$\frac{\partial A_t}{\partial t} \leq L \bullet K \bullet R(C_t) = g\,(C_{\max} - \|C_t\|) \tag{38}$$

Where L is the number of hierarchical levels (from the sum over L in previous equations), and K is the architecture-dependent constant we defined earlier. In order to satisfy the equality in

Equation 38 while ensuring necessary scaling related to computational resources, we define g as:

$$g(x) = L \cdot K \cdot \left(\frac{x}{C_{\max}} \right) \tag{39}$$

Equation 39 preserves the $L \cdot K$ scaling factor, normalizes the computational resources by $C_{\max}$, and ensures that the bound decreases in a monotonic way as computational resources are used.

b. Hierarchy

Because

$$\dim\left(span \left\{ x \in V_t : x \ = g_l \sum_{i=1}^{|V_t|} \left(\alpha(i,t) \binom{|V_t|}{i} \right) \right\} \right) \geq 0 \tag{40}$$

then

$$\dim(A_{t+1}) = dim\,(A_t) + dim\left(span \left\{ g_l \sum_{i=1}^{|V_t|} \left(\alpha(i,t) \binom{|V_t|}{i} \right) \right\} \right) \geq \dim(A_t) \tag{41}$$

Equation 40 states that the dimension of the space covered by the hierarchical transformation g_l applied to all possible token combinations must be non-negative. These equations prove that our model's semantic space grows hierarchically, adding new dimensions as it explores more complex combinations of tokens, while never losing existing dimensions.

c. Computational capacity constraints

The computational capacity condition ensures that the growth of the model's accessible state space remains bounded by available computational resources. This is formalized as:

$$\|A_{t+1} - A_t\|^2 \leq K \cdot R\,(C_t) \tag{42}$$

According to Lemma 3, we can prove the following:

$$\begin{gathered} \|A_{t+1} - A_t\|^2 = \left\| \sum_{l=1}^{L} g_l \sum_{i=1}^{|V_t|} \left(\alpha(i,t) \binom{|V_t|}{i} \right) \ \right\|_2 \leq \\ \sum_{l=1}^{L} \|g_l\|_2 \bullet \left\| \sum_{i=1}^{|V_t|} \left(\alpha(i,t) \binom{|V_t|}{i} \right) \ \right\|_2 \end{gathered} \tag{43}$$

Inequality in Equation 42 shows that the magnitude of change in the accessible space between any two time steps, $\|A_{t+1} - A_t\|^2$, cannot exceed what the system's computational resources allow, $K \cdot R\,(C_t)$. This condition is necessary because it mathematically guarantees that the model's exploration of new possibilities remains computationally feasible, preventing the system from attempting to access states that would exceed its resource capacity. In Kauffman’s terms, it guarantees that the adjacent states are possible.

5.3.4. Attention mechanism and TAP structure

The connection between attention mechanisms in language models and the combinatorial structure of TAP represents an important connection between neural computation and theoretical biology. This connection becomes apparent via a formal isomorphism, showing the natural mapping of attention operations to the combinatorial selection process of the TAP framework.

Lemma 4: Attention-TAP isomorphism. An attention space $\mathcal{A}$ forms a category where objects are attention triplets $(Q,\ K,\ V)$∈ $\mathbb{R}^d$. Morphisms are attention operations, and composition is given by sequential attention application. A TAP space $\mathcal{T}$ forms a category where objects are combinatorial sums $\sum_{i=1}^{M_t} \alpha_i \left(\frac{M_t}{i}\right)$, and morphisms are constraint-preserving transformations. Composition preserves the bounds on α_i in $[0,1]$.

Given the attention space $\mathcal{A}$:

$$\mathcal{A} = \left\{ A(Q \times K \times V) | Q, K, V \in R^d \right\} \tag{44}$$

and the TAP space $\mathcal{T}$:

$$\mathcal{T} = \left\{ \sum_{i=1}^{M_t} \alpha_i \left(\frac{M_t}{i} \right) | \alpha_i \in [0,1] \right\} \tag{45}$$

The isomorphism $\psi : \mathcal{A} \to \mathcal{T}$ has two important properties:

- $\dim(Im(\psi)) = dim\,(span\ \alpha_i)$ - dimensionality preservation-, and
- $P(x) \leq P(y) \Longleftrightarrow \psi P(x) \leq \psi P(y)$ - probability structure -.

If F is the functor that takes attention operations to their vector space representations, and G is the functor that takes TAP combinatorial selections to their probability distributions, there exists a natural transformation $\eta_a : F(a) \to G(a)$ that commutes with morphisms in both categories[98].

6. Experimental work

The main goal of this research is to validate the accuracy of our TAP framework-based equation (Equation 34) when predicting emergent properties in language models. This equation suggests that language models evolve through constrained exploration of their possibility space, driven by three main mechanisms: phase transitions in semantic space, multiplicative interaction of constraints, and path-dependent evolution. To systematically validate these theoretical predictions, we propose the following three hypotheses.

6.1. Research hypotheses

H1: Phase transitions in semantic space correlate with capability emergence. Our hypothesis is that increases in model capabilities occur through discrete phase transitions in the semantic space, rather than through gradual improvements. We should observe sudden shifts in the model's ability to handle increasingly complex tasks. Specifically, we predict that:

- The effective dimensionality of the semantic space shows sudden increases at critical points.
- These critical points correspond to the emergence of new capabilities.
- The transitions follow power-law scaling relationships characteristic of phase transitions in complex systems.
- Resource requirements (computational and context) show distinct scaling behaviors before and after the transition.

H2: Constraint interactions shape capability boundaries. We propose that model capabilities are shaped by the multiplicative interaction of three types of constraints: architectural, training, and contextual. This hypothesis predicts that:

- Performance limitations arise from the multiplicative effect of multiple constraints rather than from single bottlenecks.
- Relaxing any single constraint produces limited improvement unless other constraints are similarly relaxed.
- The impact of expanding computational resources depends on the state of other constraints.
- Models with similar total computational allocation but different constraint distributions will show distinct capability patterns.

H3: Path dependence affects problem-solving trajectories. Our third hypothesis postulates that the non-ergodic nature of language models creates significant path dependence in their problem-solving capabilities. This can be observed through:

- Different solution trajectories emerging from identical problems presented with different context orderings.
- Initial conditions (like prompt design) having persistent effects throughout the problem-solving process.

- The existence of "unreachable" solutions despite their theoretical accessibility within the model's capability space.
- Time-asymmetric behavior where forward and reverse problem-solving paths show fundamentally different characteristics.

6.2. Experimental setup

To validate our hypothesis about phase transitions in semantic space, we designed our experiments using three different language models: gpt2-xl with 1.5B parameters[8], opt with 1.3B parameters[99], and pythia with 1.4B parameters[100]. These models were selected for being open source and for their similar parameter counts but distinct architectural approaches, allowing us to separate the effects of architectural differences while controlling for model scale. gpt2-xl represents a mature architecture with established performance characteristics, while opt-1.3B and pythia-1.4B offer more recent architectural innovations but potentially less optimized training regimes.

To evaluate our hypothesis about phase transitions in semantic space, we used the high school mathematics subset of the MMLU (Massive Multitask Language Understanding) dataset, which provides a standardized set of multiple-choice questions[101]. The dataset was divided into three difficulty levels (easy, medium, hard) based on sequential ordering, with 90 questions per level. While this division method is simple, it provides a regular basis for comparing model performance across increasing task complexity. All models were evaluated using a consistent prompt format, with questions and choices formatted identically to minimize prompt-related variance. The experiments used 16-bit floating-point precision to balance computational efficiency with numerical stability. This setup allows for direct comparison of model behaviors while managing computational resources effectively.

6.2.1. Hypothesis 1: Phase transition in semantic space

Our experimental design focused on three key measurements: performance accuracy, attention entropy, and effective dimensionality of the semantic space.

Performance was measured through multiple-choice accuracy, with each model processing questions in batches of size four to optimize GPU memory use while maintaining consistent evaluation conditions. For its calculation, we used the following formula[102]:

$$A \; = \; \frac{1}{N}\sum_{i=1}^{N} 1(y_i = \hat{y}_1) \tag{46}$$

where N is the number of questions, $1(y_i = \hat{y}_1)$ is the indicator function that returns 1 when the prediction $\hat{y}_1$ matches the true answer y_i and 0 otherwise.

Attention entropy was calculated using the final layer's attention patterns, providing insights into how models distribute their focus across input tokens[103]. For its calculation, we used the following formula:

$$H(A) = -\sum_{i=1}^{N} a_i log(a_i) \tag{47}$$

where a_i are the normalized attention weights.

Finally, effective dimensionality was calculated using PCA analysis of attention patterns, finding the number of components needed to explain 90% of the variance. For calculating effective dimensionality d_{eff} we used the following formula:

$$d_{eff} = min\left\{ k : \sum_{i=1}^{k} \lambda_i / \sum_{i=1}^{k} \lambda_i \geq 0.9 \right\} \tag{48}$$

where n is the total number of dimensions - or eigenvalues[104]- in the original attention pattern space, i is the index variable for summing over eigenvalues, k is the variable we're trying to minimize that represents how many principal components are needed to explain 90% of the variance, and λ_i are the eigenvalues of the attention pattern covariance matrix[105].

6.2.2. Hypothesis 2: Constraints interaction analysis

To validate our hypothesis about constraints interactions shaping capability boundaries, we used the same three language models. These models have similar parameter counts but distinct architectural approaches, which allows us to analyze how different constraint distributions affect model capabilities while controlling for overall model scale. We also used the high school mathematics subset of the MMLU dataset. This dataset allows us to study how constraints interact across varying task complexities while providing a direct comparison with our H1 results. The constraint measurements were calculated as follows:

Architectural constraints. The architectural constraints are indeed measured using Shannon’s formula as in Equation 47 because it effectively quantifies the model's capacity to distribute attention across tokens, providing a natural way to measure how evenly the model can distribute its computational resources. The architectural constraints in language models prove primarily through the attention mechanism's capacity to distribute focus across input tokens. This distribution capacity is fundamentally limited by the model's architectural design, the number and structure of attention heads, the dimensionality of key/query vectors, and the computational paths available for information flow. Then we can compute entropy to measure architectural constraints with an equation similar to (47):

$$\beta = -\sum_{i=1}^{N} a_i log(a_i) \tag{49}$$

where a_i are the normalized attention weights. This equation captures two critical aspects of the model's architectural limitations:

a. Information processing capacity: higher entropy indicates more uniform attention distribution, suggesting the architecture can effectively process multiple inputs simultaneously.
b. Structural bottlenecks: lower entropy indicates focused attention, potentially reflecting architectural limitations in parallel processing.

Training constraints: The calculation of this constraint is computed using Equation 46. This equation is rooted in existing test theories[102], as well as in novel neural networks evaluation metrics[106]:

$$\gamma \ = \ \frac{1}{N}\sum_{i=1}^{N} 1(y_i = \hat{y}_1)c_i \tag{50}$$

However, its theoretical interpretation and application change significantly. Equation 46 evaluates the accuracy of the raw performance, while Equation 50 quantifies the limitations set by the system's training patterns. The study of information bottlenecks in deep learning parallels this approach[107]. Equation 50 captures training constraints since it recognizes pattern limits as true predictions $(y_i = \hat{y}_1)$, and shows learned patterns reflecting the model's acquired inductive biases[108]. The added confidence term c_i indicates the strength of learned associations inspired by the work in neural network uncertainty quantification[109]. Finally, the normalized sum represents the overall training pattern constraints following statistical learning theory[110].

Contextual constraints. According to Achille & Soatto[111], contextual constraints reveal themselves through specific mechanisms that shape how information flows through the network. The cited paper discusses several possible contextual constraint mechanisms such as sequential dependency or context bottleneck effect. We can write our constraint as

$$\delta = std\left(attention\_weights[: windowsize]\right) \tag{51}$$

Metric $std(attention\_weights)$ reflects how consistently the model uses context (which is related to sequential dependency), shows how variable attention is distributed (which is related to the context bottleneck effect), and it's easy to compute during training. This measurement aligns with our theoretical framework, which postulates that contextual constraints emerge from the interaction between the model architecture (attention mechanism), training objectives (task sufficiency), and information bottleneck effects (layer stacking).

All three constraints directly affect the system's ability to explore its adjacent possible states following the original TAP framework:

$$P(true|training) \propto (1-\beta)(1-\gamma)(1-\delta)P(state|architecture) \tag{52}$$

6.2.3. Hypothesis 3: Path dependence influences problem-solving trajectories

Our experimental design for analyzing path dependence focused on four complementary measurements, capturing different features of solution trajectory changes. As test data, we used the high school mathematics subset of the MMLU dataset[101] with 30 questions to ensure statistical robustness. The four metrics are the following:

a) Building on trajectory analysis methods from stochastic processes[14], we compared solution paths under normal and shuffled input conditions, measuring step count variations, consistency differences, and directness disparities. We used the expression

$$\Delta_{path} = |M_{normal} - M_{shuffled}| \tag{53}$$

where M represents metrics including the number of steps, path consistency, and solution directness. Solution paths were analyzed both in their original order and with shuffled choice presentations to assess path dependence effects.

b) Consistency through the model's hidden state representations is measured using cosine similarity between consecutive solution steps. We extracted the final layer's hidden states for each solution step and computed mean embeddings across the sequence length dimension. Finally, we calculated cosine similarity between consecutive step embeddings. This approach follows methods established for analyzing neural network internal representations[37]. We used the following equation:

$$C = \frac{1}{n-1} \sum_{i=1}^{n-1} \cos\left(s_i, s_{i+1}\right) \tag{54}$$

where s_i represents the semantic embedding of step i and n is the total number of steps.

c) We quantified solution directness through a combination of step count and revision detection, where revisions were identified through specific linguistic markers (e.g., “actually”, “instead”, “correction”) following approaches from solution path analysis[4]. To compute the solution directness, we use the following equation:

$$D = \frac{1}{1 + |steps|} + \frac{1}{1 + |revisions|} \tag{55}$$

where $|steps|$ is the total number of solution steps, and $|revisions|$ counts backtracking instances.

d) Finally, the step length variations were computed with the following equation:

$$L_{diff} \ = \ L_{normal} - L_{shuffled} \tag{56}$$

where L is the average step length under normal and shuffled conditions.

This experimental design is rooted in existing methodologies for analyzing stochastic systems, which we adapted to the unique context of large language model evaluation. The approach aligns with previous work on analyzing non-ergodic behavior in complex systems[18]. The key limitation of this setup is potential noise in path difference measurements for complex problems, addressed through multiple trials and robust statistical controls[112]. This approach enables systematic comparison of path dependence effects across different model architectures while maintaining statistical robustness and measurement reliability. Each metric captures a distinct aspect of path dependence, from direct trajectory differences to more subtle variations in solution characteristics. Following standard practices in language model evaluation, we used temperature sampling for generation.

7. Experimental results

7.1. Phase Transitions in Semantic Space (H1)

The first hypothesis proposes a correlation between the emergence of capabilities in language models and phase transitions in semantic space. The experimental results support this prediction, although there are significant distinctions in the occurrence of these transitions. These results may explain why models cannot perpetually expand their semantic complexity as well as the predictable impact of resource allocation on model performance. Finally, it proves the inherent limitations of model scaling based on available resources.

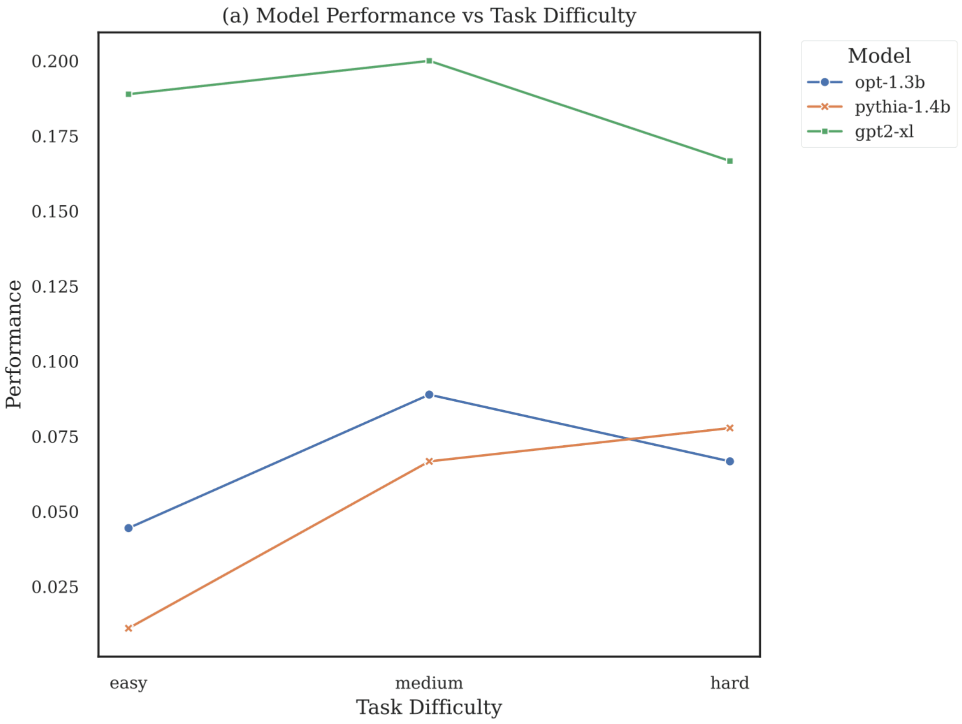

Figure 1. Model performance and task difficulty

Figure 1 illustrates how both gpt2-xl and opt-1.3B exhibit inverted U-shaped performance trajectories, with a peak at medium difficulty at different performance levels (gpt2-xl peak: 0.200, opt-1.3B peak: 0.089). In contrast, pythia-1.4B shows a different pattern with a sharp initial increase from easy to medium, followed by continued improvement, unveiling a different form of phase transition in its capability space.

The semantic space analysis in Figure 2 shows how these transitions correlate with changes in the models' operational regime. gpt2-xl (green) operates in a distinctive regime characterized by a high attention entropy (~1.7) and lower effective dimensionality (8-10 dimensions). This suggests the model distributes attention broadly but in a more compact semantic space. In contrast, opt-1.3B (in orange) and pythia-1.4B (in blue) show lower attention entropy (~1.3-1.4) and higher effective dimensionality (14-20 dimensions). This could indicate more focused attention patterns but across a larger semantic space. The size of the dots represents model performance, providing additional

doi.org/10.32388/00JQN6

insight into how these different regimes relate to model capabilities. This visualization reveals a fundamental trade-off in language model design: models can either operate with high entropy in a compact space (like gpt2-xl) or with more focused attention across a larger dimensional space (like opt-1.3B and pythia-1.4B). These distinct operational regimes suggest different strategies for managing the complexity of language processing, with implications for how phase transitions in capabilities emerge in each architecture type.

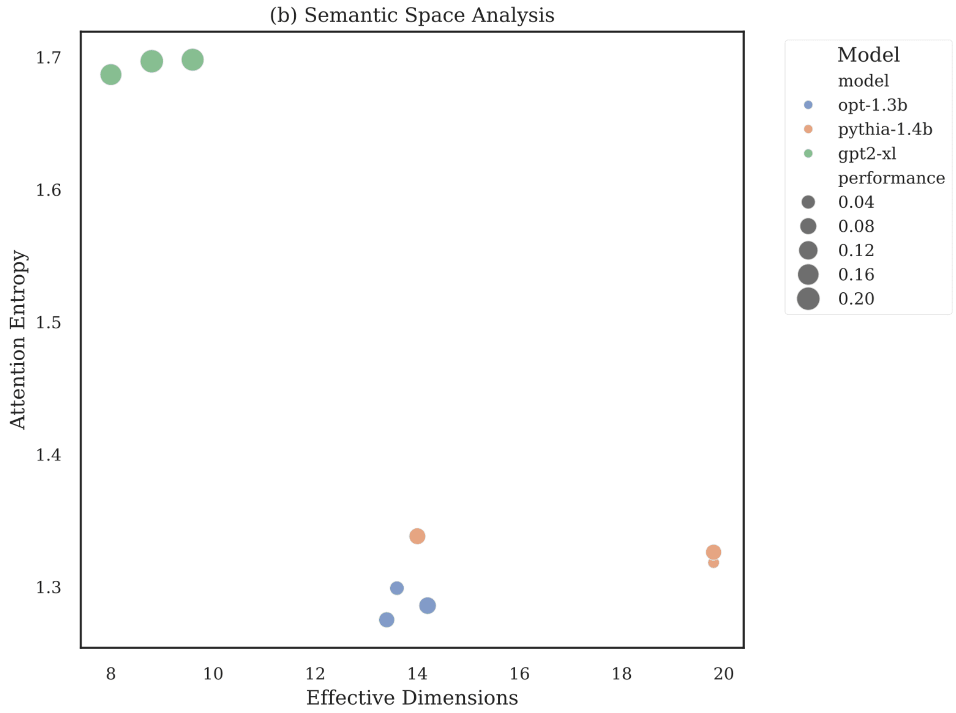

Figure 2. Semantic space analysis.

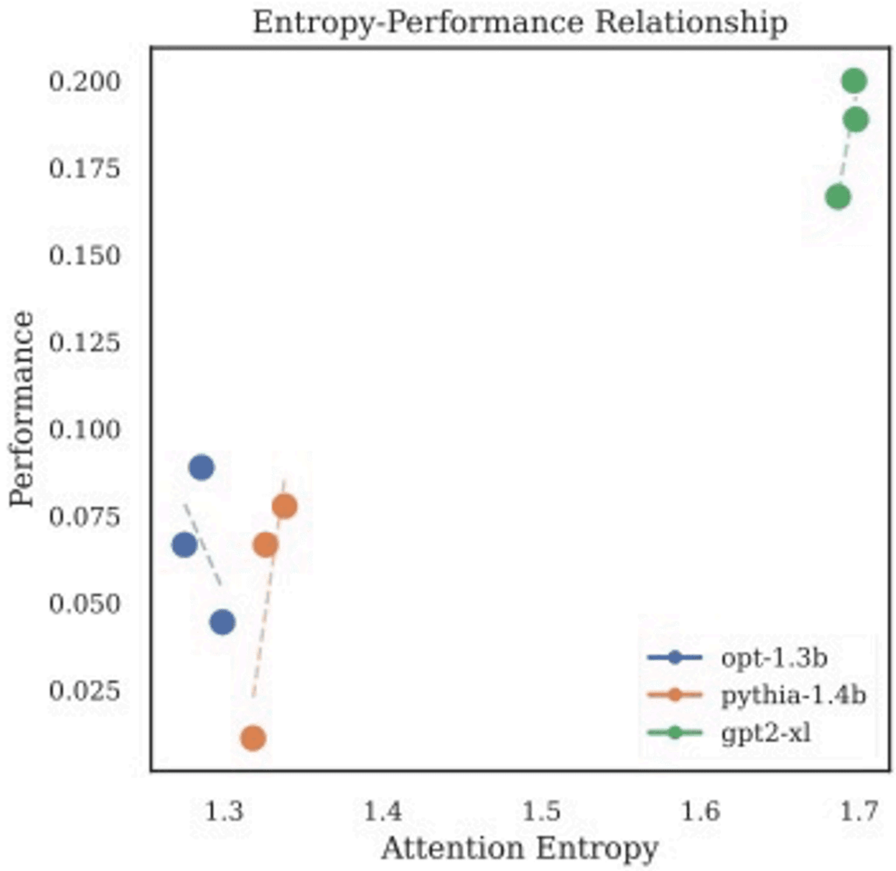

Figure 3. Correlation between performance and entropy.

In Figure 3, strong correlations between performance and entropy for each model (gpt2-xl: 0.910, pythia-1.4B: 0.883) suggest that transitions follow organized patterns rather than random fluctuations. The varying stability metrics (gpt2-xl: 0.925, opt-1.3B: 0.728, pythia-1.4B: 0.438) further suggest that different models show different types of phase transitions, from smooth progressions to sharp capability shifts.

7.2. *Constraint interactions shape capability boundaries (H2)*

Our second hypothesis considers that model capabilities are shaped by multiplicative interactions between architectural, training, and contextual constraints. The experimental results can be seen in the following tables.

Performance stability metrics shown in Table 1 suggest different behavior patterns across architectures. Model gpt2-xl shows higher overall performance (mean=2.3477) while maintaining moderate stability (CV=3.26%). opt-1.3b shows the most stable performance (CV=2.68%) but lower absolute performance (mean=1.8089), and pythia-1.4b has the highest variability (CV=6.97%), suggesting less robust constraint management. The stability patterns indicate that architectural differences significantly influence how models manage constraint trade-offs, with gpt2-xl achieving the best balance between performance and stability.

Model	Std	CV(%)	Range	Min	Max	Mean
op-1.3B	0.0484	2.68	0.096	1.7639	1.8602	1.8089
pythia-1.4B	0.1281	6.97	0.2471	1.7340	1.9811	1.8380
gpt2-xl	0.0765	3.26	0.1461	2.2616	2.4077	2.3477

Table 1. Performance stability metrics.

Table 2 shows the analysis of constraint interactions across different difficulty levels. Architectural constraints show a systematic decrease with difficulty (from 6.2444 to 5.9117). The highest effect in easy tasks suggests more efficient architectural utilization, and the significant drop in hard tasks points to architectural strain. Training Constraints show non-monotonic behavior (from 0.1300 to 0.0000 and to 0.1261) with a valley at medium difficulty, suggesting a critical transition point. Recovery in hard tasks suggests adaptive training dynamics. Contextual Constraints reveal a slight increase with difficulty level (from 0.0831 to 0.0935), being the most stable among all constraints. It suggests increasing reliance on context for complex tasks. The weighted constraints combination ("Performance" column in Table 2) shows peak performance in easy tasks (2.0829), then stabilizes around 1.94-1.97 for medium/hard tasks. These results provide evidence that constraint interactions adapt dynamically to task complexity. The systematic decrease in architectural effects with increasing difficulty supports our hypothesis about adaptive constraint dynamics.

doi.org/10.32388/00JQN6

	β	γ	δ	**Performance**
Easy	6.2444	0.1300	0.0831	2.0829
Medium	6.2380	0.0000	0.0826	1.9414
Hard	5.9117	0.12611	0.0935	1.9702

Table 2. Analysis of constraints across difficulty levels. The performance column represents a weighted combination of architectural, training, and contextual constraints effects (30% β, 40% γ, and 30% δ), plus an additional 10% contribution from their interaction. β are the architectural constraints, γ are training data constraints, and δ are contextual constraints.

7.2.1. Constraints analysis

Analysis of normalized constraint values in Table 3 shows several striking patterns. For example, architectural constraints remain consistent across models (from 0.6348 to 0.6409), revealing a common architectural bottleneck. This suggests that despite different architectures, all three models hit similar fundamental limitations in how they can process information. Training constraints show higher variance (from 0.3333 to 0.5193), with the pythia-1.4b model having the lowest normalized training constraint impact. This variation suggests that models learn and utilize training data differently. Contextual constraints show a clear progression (from 0.3350 to 0.4054). The model gpt2-xl shows a bigger impact of context constraints, which is consistent with prior entropy analyses indicating a wider attention distribution for gpt2-xl.

Model	β	γ	δ
op-1.3B	0.6348	0.5000	0.3350
pythia-1.4B	0.6409	0.3333	0.3562
gpt2-xl	0.6378	0.5193	0.4054

Table 3. Normalized constraint values. β are the architectural constraints, γ are the training data constraints, and δ are the contextual constraints.

In Table 4, we can see how training constraints consistently show the highest relative importance (from 0.7506 to 1.4952), pointing to their primary role in shaping the model's performance. Architectural constraints represent model-specific relative importance, with the opt-1.3b (0.8325) and gpt2-xl (0.7314) showing similar dynamics. Contextual constraints show lower but consistent relative importance (from 0.1581 to 0.5223). The high R^2 score values confirm the significance of these results.

Model	β imp.	γ imp.	δ imp.	R^2 score
op-1.3B	0.8325	1.4591	0.5223	0.9634
pythia-1.4B	0.1645	0.7506	0.1581	0.9984
gpt2-xl	0.7314	1.4952	0.1667	0.9513

Table 4. Relative constraints importance. β are the architectural constraints, γ are the training data constraints, and δ are the contextual constraints.

Model	β threshold	γ threshold
opt-1.3B	5.72	0.11
pythia-1.4B	5.80	0.11
gpt2-xl	7.45	0.06

Table 5. Architectural constraints β, and training constraints γ threshold.

Table 5 shows different thresholds in the effects of architectural and training constraints for all models. This supports our hypothesis that capability boundaries are influenced by the interactions of constraints. The gpt2-xl model has a higher architectural threshold in comparison to other models. This indicates improved performance, implying that architectural constraints significantly influence model capabilities.

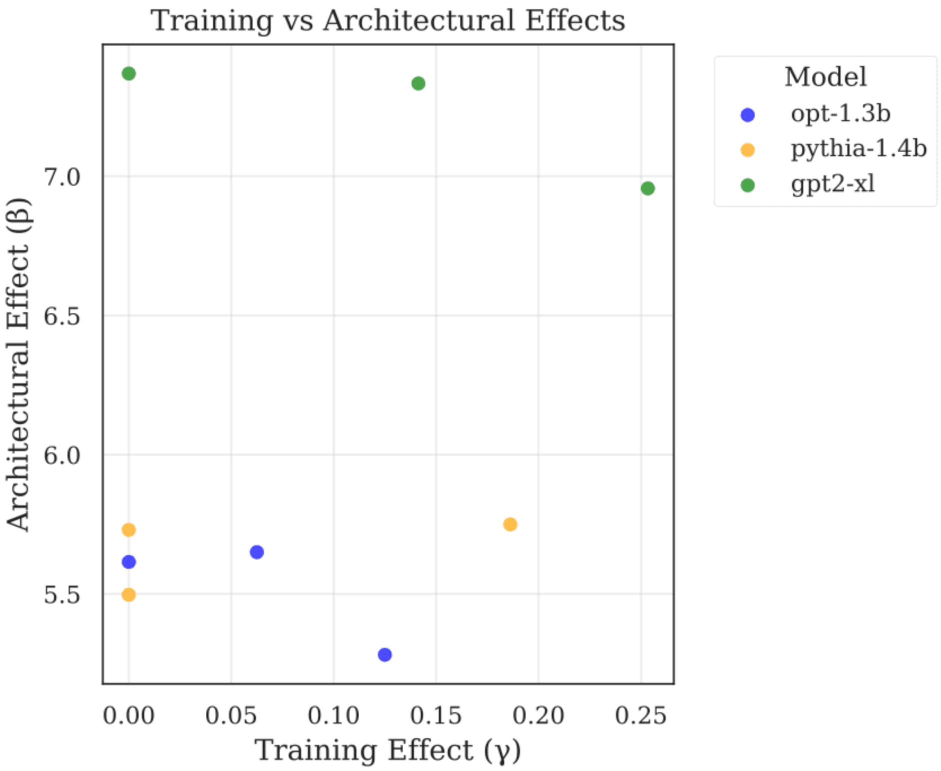

Figure 4. Training vs architectural constraints effects.

The correlation between training and architectural constraints shown in Figure 4 reveals uncorrelated dynamics. gpt2-xl (green dots) shows consistently high architectural effects (~7.0-7.5) regardless of training effects. opt-1.3B (blue dots) and pythia-1.4B (orange dots) cluster together at lower architectural effects (~5.5-6.0). This clear separation suggests fundamentally different operational regimes between gpt2-XL and the other models. Training effects range from 0.0 to 0.25 across all models with no clear pattern or correlation between training and architectural effects. Points are scattered horizontally, suggesting training effects vary independently of architectural constraints. The lack of correlation suggests architectural and training constraints operate independently. While gpt2-xl maintains higher architectural effects despite training variations, opt-1.3B and pythia-1.4B show similar architectural behaviors across different training effects.

Training and contextual constraints correlations in Figure 5 vary significantly by model. The model gpt2-xl shows a precise positive correlation (R^2=1.0 with p-value=0.000), while the pythia-1.4b and opt-1.3b models offer less significant correlations (R^2=-0.866 with p-value=0.333, and R^2=0.500 with p-value=0.667 respectively). Figure 6 illustrates how architectural and contextual constraints correlations show a precise negative correlation across all models (R^2=-1.0 with p-value=0.000), indicating a fundamental trade-off between both constraints.

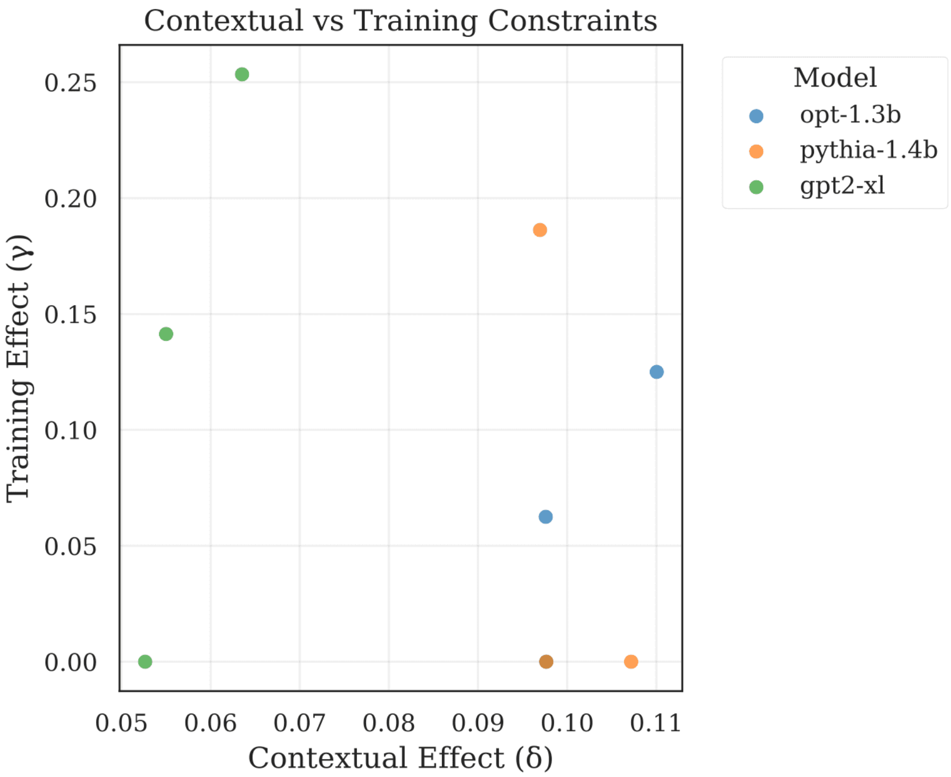

Figure 5. Training vs contextual constraints effects.

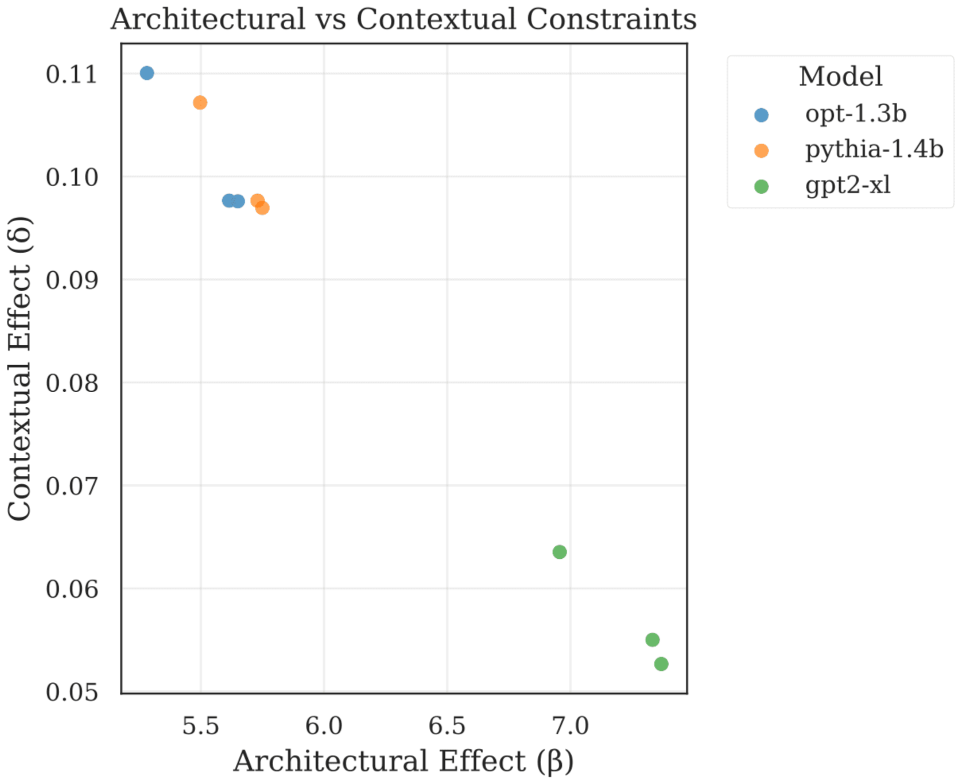

Figure 6. Architectural vs contextual constraints effects.

Figure 7 shows an interesting distribution of constraint effects across three different models, and its results connect directly to our theoretical framework in several important ways. The influence of architectural constraints in all three models compared to contextual and training constraints (near 0) aligns with our theoretical prediction that architectural constraints are fixed at initialization and fundamentally limit the model's capability space. Particularly interesting is that gpt2-xl shows the highest architectural constraint, suggesting its architecture more strictly bounds its possibility space. These results reflect our theoretical framework's emphasis on architectural constraints through $R(C_t)$. However, we need additional independent evidence to validate whether this emphasis accurately represents the real dynamics of language models.

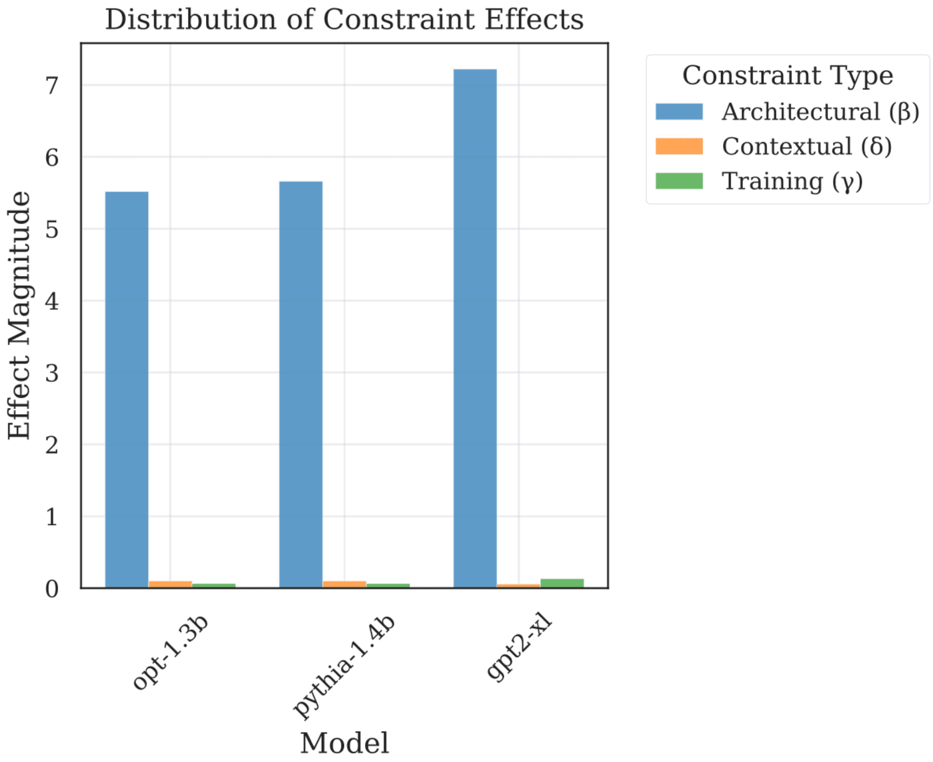

Figure 7. Constraints distribution for different models.

Figure 8 shows the relationship between combined constraint effects ($\beta \times \gamma \times \delta$) and model performance, revealing critical phase transitions in the constraint space. The plot demonstrates distinct model behaviors and clear transition points. Model gpt2-xl has the highest performance (from 2.2 to 2.4), maintaining stable performance across a wider range of combined constraint values (from 0.06 to 0.12). In contrast, the opt-1.3b and pythia-1.4b models operate in a lower performance region (from 1.7 to 2.0) and show more abrupt transitions. Each model demonstrates a distinct critical threshold (marked with vertical dashed lines): gpt2-xl at 0.10, pythia-1.4b at 0.08, and opt-1.3b at 0.07. The pre-transition region (red shaded area) represents a phase where models operate below their optimal constraint balance, while the post-transition region (blue shaded area) indicates where models achieve better constraint integration. These thresholds mark points where models transition from lower to higher performance states. Higher thresholds correlate with better overall performance. For example, model gpt2-xl not only has the highest threshold but also maintains more consistent performance in the post-transition region. This suggests that its architecture achieves a more robust balance of constraints. The sharp performance shifts at these thresholds indicate the non-linear nature of constraint interactions and the existence of critical points where model behavior has a symmetry breaking.

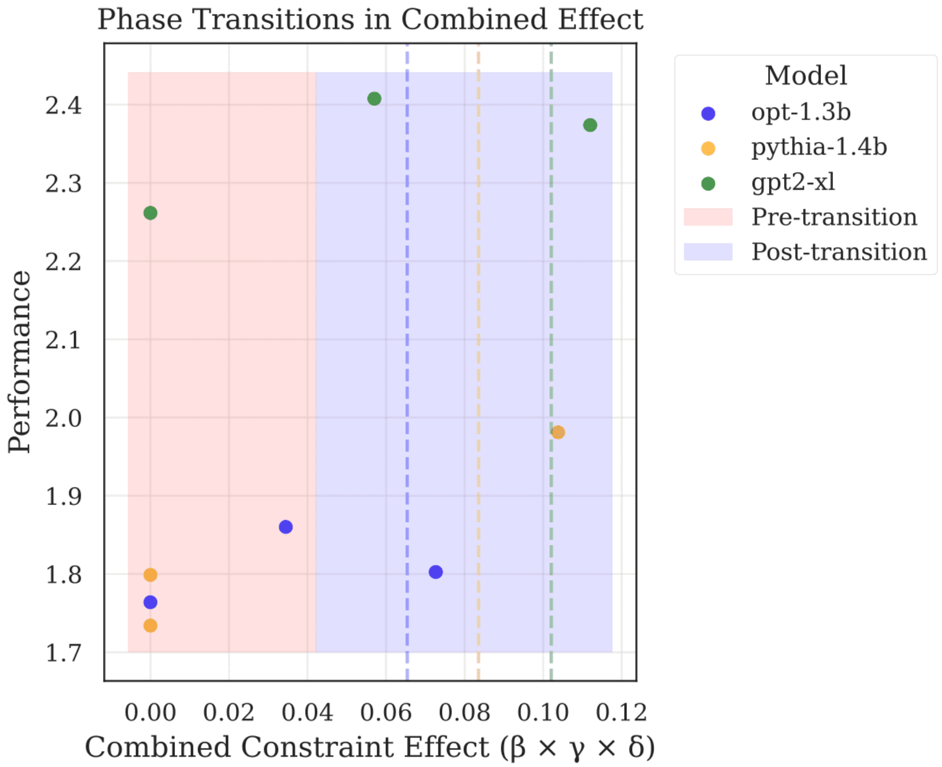

Figure 8. Phase transitions for combined constraints across different models.

Our experimental results provide strong support for hypothesis 2 through various pieces of evidence. First, the performance-stability analysis shows how different architectures manage constraints distinctly, with gpt2-xl achieving optimal balance while maintaining high performance. Second, the constraint interaction analysis across difficulty levels reveals dynamic adaptation, evidenced by the systematic decrease in architectural effects and the architectural/contextual constraints ratio demonstrating how constraints adapt to task complexity. Third, the correlation analysis reveals fundamental trade-offs, particularly the perfect negative correlation between architectural and contextual constraints. Finally, the phase transition analysis reveals clear threshold effects in both architectural impact and combined constraint interactions, where each model shows distinct but related critical points. These transitions and model-specific thresholds demonstrate that performance emerges from multiplicative interactions between constraints rather than simple additive effects, conclusively supporting our hypothesis that model capabilities are shaped by complex interactions between architectural, training), and contextual constraints.

7.3. Constraint interactions shape capability boundaries (H3)

Results for Hypothesis 3 can be seen in the following figures. Figure 9 shows the path differences across models. Model opt-1.3b shows the highest number of step differences (0.447), suggesting the strongest path dependence in solution approach. Models pythia-1.4b and gpt2-xl show moderate step differences (0.347 and 0.287 respectively), suggesting more stable solution strategies despite input order.

Figure 10 compares the step length variability across models. Pythia-1.4b shows the largest step length difference (15.957), indicating high sensitivity to input ordering in terms of solution verbosity. Model opt-1.3b shows moderate step length variation (11.941), and model gpt2-xl shows high but consistent step length differences (13.819). Figure 10 shows the consistency vs. directness analysis dynamics, not revealing any important pattern to be considered.

In Figure 11, we observe that all models have similar directness differences (0.020-0.021), suggesting comparable efficiency in reaching solutions regardless of the path. Consistency differences vary more significantly, with model opt-1.3b having the highest value (0.067) and gpt2-xl the lowest (0.032). The tight clustering of directness differences despite varying consistency differences suggests that models maintain solution efficiency even when following different paths.

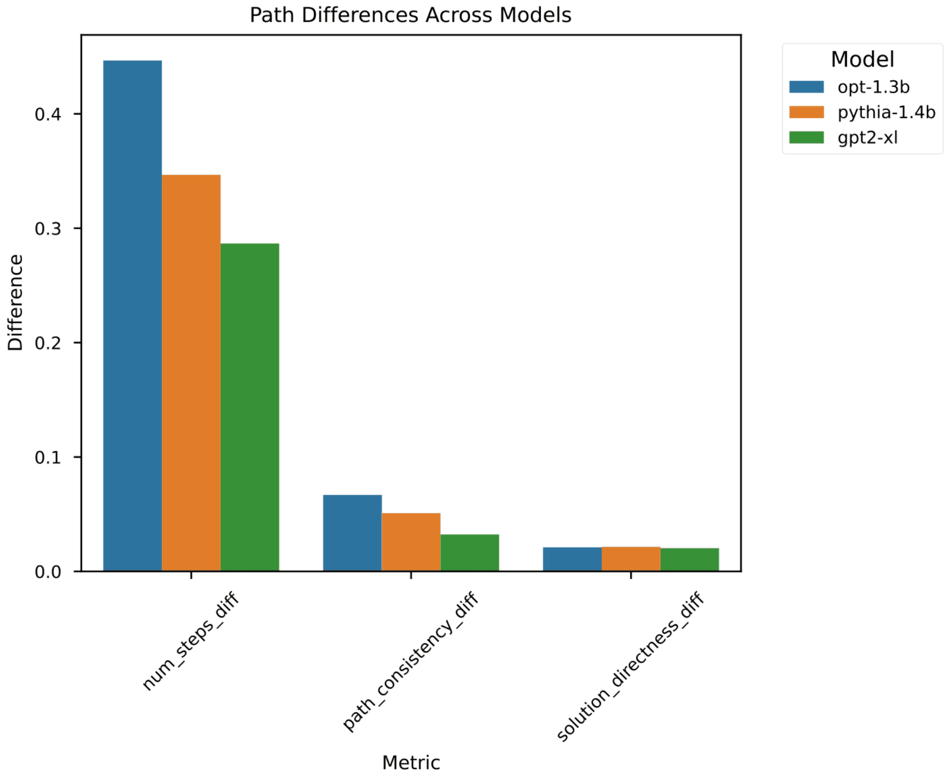

Figure 9. Path differences across models.

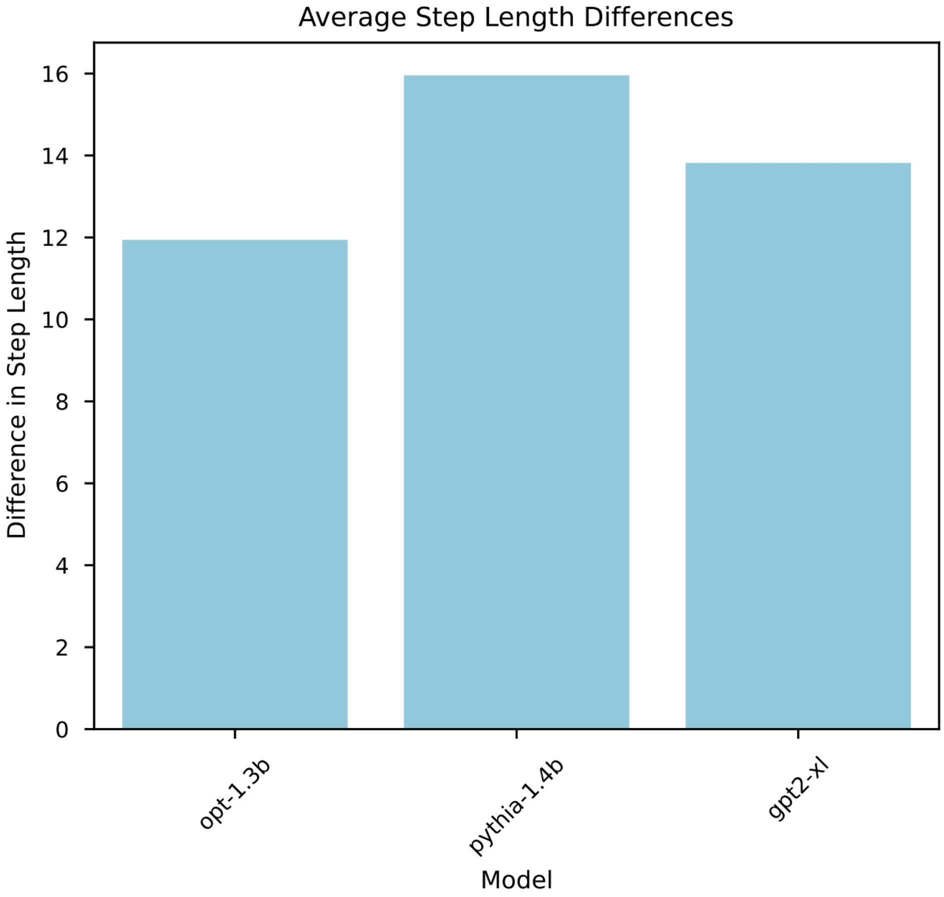

Figure 10. Step length analysis across models.

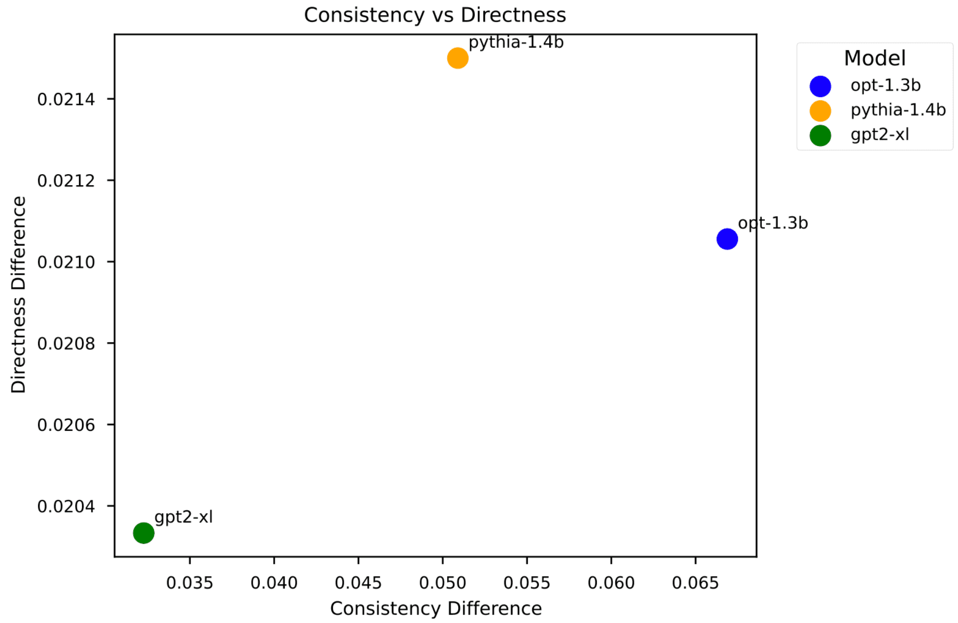

Figure 11. Path differences across models.

Figure 12 shows a comparison of different characteristics across the models. The model gpt2-xl suggests the most balanced performance across different metrics, showing moderate differences in step length and the lowest variation in consistency. The opt-1.3b model demonstrates significant path dependence while ensuring consistent solution directness. Model pythia-1.4b has the greatest variability in step length while also showing moderate levels of consistency and directness metrics.

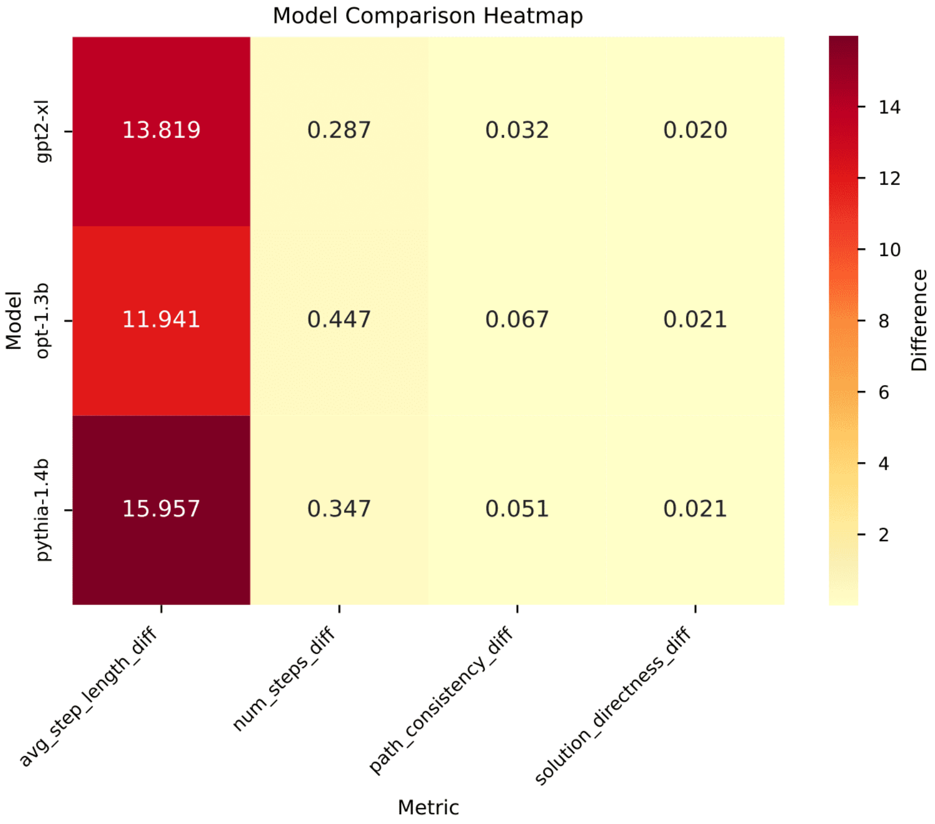

Figure 12. Path differences across models.

These results provide support for H3, suggesting that path dependence significantly affects problem-solving trajectories across all models. While all models exhibit path dependence, they use different strategies to maintain solution quality: gpt2-xl prioritizes consistency, opt-1.3b balances between path exploration and directness, and pythia-1.4b shows high adaptability in solution length while maintaining solution quality. The observed variations in path-dependent strategies align with our theoretical framework's predictions about non-ergodic exploration in constrained possibility spaces. gpt2-xl's higher consistency (lowest consistency difference 0.032) reflects its architectural capacity for parallel processing through multiple attention heads[35], enabling stable representation maintenance across different solution paths. Model opt-1.3B's balanced approach, with moderate step differences (0.447), emerges from its enhanced layer connectivity that facilitates diversified path exploration while maintaining solution coherence[10]. Pythia-1.4B shows high adaptability in solution length (15.957) while maintaining directness (0.021), indicating that position-aware processing through scaled rotary embeddings enables flexible path exploration within semantic constraints[100].

These architectural differences manifest in our TAP equation through the hierarchical function g_l, where each model's specific attention mechanisms and layer connectivity patterns create distinct mappings between token combinations and semantic space. The varying magnitudes of path differences across models quantify how architectural constraints shape the adjacent possible states accessible during problem-solving, supporting our framework's prediction that capability emergence follows architecture-dependent trajectories through the possibility space.

8. Discussion

Our experimental results provide strong support for the application of TAP theory to understanding emergent capabilities in large language models, while also revealing important nuances in how these systems navigate their possibility spaces. The findings validate our proposed resource-bounded TAP equation and demonstrate that language models exhibit behavior patterns consistent with complex biological systems, particularly in terms of phase transitions, constraint interactions, and path dependence.

8.1. Theoretical framework validation

The experimental results provide substantial support for our theoretical framework across several important areas.

First, the observation of clear phase transitions in semantic space aligns with TAP theory's prediction of discrete shifts in capability as systems explore their adjacent possible states. The inverted U-shaped performance curves observed in gpt2-xl and opt-1.3B suggest that these transitions are not simply cumulative improvements but rather represent fundamental reorganizations of the models' operational regimes. This behavior parallels Kauffman's description of biological systems transitioning between distinct organizational states.

Second, the multiplicative interaction of constraints observed in our experiments supports our theoretical decision to model constraint interactions multiplicatively instead of additively in our TAP equation. The observed negative correlation between architectural and contextual constraints indicates that these constraints operate as fundamentally interconnected parameters, similar to the combined constraints found in biological systems within metabolic networks.

Third, the strong path dependence observed across all models confirms the non-ergodic nature of these systems, a key assumption in our theoretical framework. The varying consistency differences between models while maintaining similar directness metrics suggest that, like in biological systems, language models explore their possibility spaces through restricted but efficient pathways.

8.2. Emergence and phase transitions

Recent empirical research has provided specific evidence for emergence patterns in language models. Wei et al.[2] documented discontinuous improvements across 23 different capabilities in PaLM, finding that abilities like multi-step reasoning emerged suddenly at certain model scales rather than improving gradually. Analogously, Ganguli et al.[1] analyzed the predictability of model capabilities during scaling, differentiating between gradual improvements and unexpected behavioral changes. Recent work by[113] explored the nature of emergent abilities, developing more rigorous methods for categorizing between basic emergence and scaling effects. These works provide actual evidence for non-random patterns in capability emergence, setting the stage for our analysis of the diverse behavior patterns observed across models.

In our research, the diverse behavior patterns observed across models - from gpt2-xl's balanced performance to pythia-1.4B's continued improvement pattern - suggest that emergence can appear through different mechanisms depending on architectural choices and constraint distributions. This observation aligns with Holland[58] characterization of emergence in complex adaptive systems, where global patterns arise from local interactions under varying constraint conditions. The correlation between performance and entropy suggests that these transitions follow organized patterns rather than random fluctuations, relating to Prigogine's work on dissipative structures where order emerges from the interaction between system dynamics and environmental constraints[45]. The systematized nature of transitions we observed supports our theoretical framework's prediction that capability emergence follows constrained exploration paths rather than random search. This aligns with Haken's synergetics theory[60], which describes how collective behavior emerges through self-organization under constraints. Our findings follows the theory of self-organized criticality systems[114][76][65], as models appear to naturally evolve toward critical states where new capabilities emerge. The architectural dependency of emergence patterns confirm a hierarchical organization in complex systems, where different architectural configurations lead to distinct emergent properties[80]. This systematized path-dependent nature found in emergence in language models provides a connection between deterministic phase transitions in physical systems and the more complex emergence patterns seen in biological systems, suggesting a new category of emergent phenomena in artificial intelligence systems that require further theoretical investigation.

While previous research proved the existence of emergent capabilities in language models, our work provides a theoretical framework that explains why and how these capabilities emerge through using non-ergodic dynamics and the adjacent possible theory. By proving that language models operate as non-ergodic systems and demonstrating how their capability emergence is shaped by multiplicative constraint interactions, we move beyond descriptive observations to a mechanistic understanding of emergence.

8.3. Constraint dynamics, path dependence and non-ergodicity

The varying stability metrics across models suggest that different architectures create distinct hierarchical organizations of constraints, affecting how capabilities emerge The systematic decrease in architectural constraints with increasing task difficulty indicates that models adaptively redistribute their computational resources as tasks become more complex, supporting our resource-bounded formulation of the TAP equation. The non-monotonic behavior of training suggests that learned patterns play a complex role in shaping the adjacent possible space, similar to how biological systems' past adaptations influence their future possibilities.

The path dependence results provide strong evidence for the non-ergodic nature of language models, but with important qualifications. The variation in step differences across models suggests that architectural choices can significantly influence the degree of path dependence. This finding has important implications for our theoretical framework, suggesting that while all models operate in non-ergodic regimes, the strength of historical dependence can be modulated through architectural design. This parallels biological systems where different organizational structures can lead to varying degrees of historical contingency.

We have seen that models used in our experiments adopt distinct strategies to maintain solution quality. These strategies could be related to their architectures. For example, gpt2-xl's consistency priority can be related to its larger number of attention heads (25 heads per layer) and relatively smaller head dimension, enabling better parallel processing of information[8]. This architectural choice explains its observed preference for consistency, as multiple attention heads can maintain stable representations across different solution paths.

Model opt-1.3B's balanced approach can be related to the model’s additional skip connections and modified layer normalization positions compared to standard transformer architectures[99]. These architectural features facilitate information flow between different layers, explaining its balanced approach between path exploration and directness. The higher step differences and maintained directness metrics suggest that the enhanced layer connectivity enables the model to explore different paths while maintaining solution coherence.

Pythia's architecture incorporates scaled rotary embeddings and modified attention patterns that enhance its position-aware processing[100]. This architectural choice explains its observed high adaptability in solution length while maintaining quality. The largest step length differences and consistent solution quality metrics align with its enhanced position-aware processing capabilities.

These architectural-behavioral correlations suggest a generalizable principle: the distribution and structure of attention mechanisms fundamentally shape how models navigate their possibility spaces. Models with more parallel processing capabilities tend toward consistency-focused strategies, while those with enhanced layer connectivity or position-aware processing enable more flexible exploration strategies.

This finding has significant implications for our theoretical framework, suggesting that although all models operate in non-ergodic regimes, both the extent and type of historical dependence can be adapted through architectural design. This is analogous to biological systems, where diverse organizational structures may result in differing levels of historical contingency. The capacity to predict path dependence strategies based on architectural features indicates that our TAP framework captures the fundamental principles about the influence of system structure on the exploration of possibility space.

9. Practical implications and future directions

Our theoretical framework and experimental findings could have significant implications for AI development while suggesting important directions for future research. In the domain of model explainability, the TAP framework provides novel tools for understanding model behavior through constraint interactions and phase transitions. Recent research on emergence in large language models[115][116][113][2] suggests that capability shifts can be predicted through careful monitoring of model behavior, aligning with our observations of clear thresholds in capability emergence. These insights complement recent advances in mechanistic interpretability[117][118][119][120] and neural network interpretation[37][121].

The resource-bounded TAP equation (Equation 34) provides specific guidance for designing new model architectures. The multiplicative nature of constraint interactions suggests that architectural improvements should focus on the balanced enhancement of all constraints rather than optimizing single components. This understanding leads to several important architectural innovations. Attention mechanisms could be designed to maintain consistent entropy across different operational regimes, while layer connectivity patterns could facilitate both information preservation and flexible path exploration. Position-aware processing could be integrated throughout the architecture to enable adaptive context utilization. These principles extend to dynamic routing mechanisms that enable flexible path exploration and adaptive connectivity patterns supporting multiple solution trajectories.

The framework has particular relevance for AI alignment, where new research reports the importance of understanding how models internalize training objectives[122]. The non-ergodic nature of language models suggests that alignment strategies must account for path dependence, while the multiplicative nature of constraint interactions indicates that controlling multiple constraints simultaneously might be more effective than focusing on individual ones. This aligns with recent theoretical work on multi-constraint optimization in AI systems[123] and phase-aware training methods[124].

Our framework enables deliberate design for specific phase transitions in model development. Architectural features can be tuned to target desired capability emergence points, while resource allocation can be optimized based on predicted transition thresholds. Critical points in semantic space expansion can be engineered through careful constraint balance. This approach suggests that AGI capabilities might emerge through discrete shifts rather than continuous improvement. Our findings indicate that an intentional strategy for specific phase transitions in AGI development might bring greater advantages than solely depending on scaling approaches. New research on architectural innovation[70] and compute-optimal scaling[125] supports this insight.

The theoretical framework aligns with LeCun[22] approach to autonomous machine intelligence. New implementations of H-JEPA[126] test how hierarchical organizations might naturally emerge from constraint interactions. The observed path dependence in problem-solving strategies suggests that world model formation follows similar constrained exploration patterns, supporting recent advances in predictive modeling[127][128].

Likely, several promising research directions emerge from our work. The theoretical framework can be extended to develop more precise mathematical models of constraint interaction dynamics and investigate relationships between phase transitions and optimization landscapes. Practical applications include the development of real-time analysis tools based on the TAP framework and the implementation of phase-aware training algorithms. The framework also suggests rich opportunities for interdisciplinary research, particularly in exploring parallels between AI and biological learning systems through the TAP lens.

Resource management in future architectures will require built-in phase transition monitoring capabilities and adaptive allocation based on capability emergence patterns. These systems should incorporate flexible computational pathways supporting diverse problem-solving strategies while maintaining balanced resource utilization. Such architectural innovations, guided by our theoretical framework, could lead to more efficient and capable language models that exhibit more controlled and predictable emergence of capabilities.

This work opens new avenues for research while providing practical tools for immediate application in AI development. The integration of constraint-aware design principles with phase transition engineering and non-ergodic architecture principles offers a comprehensive approach to advancing AI systems. Through careful application of these principles, we can work toward developing more robust, interpretable, and capable AI systems that exhibit predictable and controllable emergence of capabilities.

Our proposal in this paper is based on Stuart Kauffman's theory of the adjacent possible. Kauffman introduced other ideas that could be very useful in the field of artificial intelligence. Future research could explore how autonomous agents in language models maximize their average rate of exploration of adjacent possible states while preserving coherent functionality, similar to capability expansion in biological systems. A deeper understanding of molecular autonomous agents' principles could provide new insights into how language models balance exploration of novel states with the maintenance of existing capabilities, particularly in continual learning scenarios. Kauffman's NK fitness landscape model[5] could provide a robust framework for understanding how language models navigate their possibility spaces through the interaction of N components (architectural elements, attention mechanisms, layer connectivity) and K epistatic interactions (constraint relationships). Future research could further explore how the roughness of these landscapes, defined by the degree of interdependence between components[129], influences the emergence of capabilities and the stability of model behavior. This perspective suggests studying how different architectural choices create varying degrees of landscape ruggedness, potentially explaining why some models exhibit more robust capability emergence while others show fragility or unpredictability. Adaptive walks on correlated fitness landscapes[129] could further provide novel training strategies that effectively balance exploration of promising regions with exploitation of established solutions.

References

1. a, b *Ganguli D, Hernandez D, Lovitt L, Askell A, Bai Y, Chen A, Conerly T, Dassarma N, Drain D, Elhage N (2022). "Predictability and surprise in large generative models". Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency: 1747–1764.*
2. a, b, c, d, e, f, g, h, i *Wei J, Tay Y, Bommasani R, Raffel C, Zoph B, Borgeaud S, Yogatama D, Bosma M, Zhou D, Metzler D. (2022). Emergent abilities of large language models. ArXiv Preprint ArXiv:2206.07682.*
3. a, b, c, d, e, f, g *Press O, Smith NA, Lewis M (2021). "Train short, test long: Attention with linear biases enables input length extrapolation". ArXiv Preprint ArXiv:2108.12409.*
4. a, b, c *Levine Y, Dalmedigos I, Ram O, Zeldes Y, Jannai D, Muhlgay D, Osin Y, Lieber O, Lenz B, Shalev-Shwartz S (2022). Standing on the shoulders of giant frozen language models. ArXiv Preprint ArXiv:2204.10019.*
5. a, b, c, d, e, f, g, h, i *Kauffman SA (2000). Investigations. Oxford University Press.*
6. ^ *Chowdhery A, Narang S, Devlin J, Bosma M, Mishra G, Roberts A, Barham P, Chung HW, Sutton C, Gehrmann S (2023). "Palm: Scaling language modeling with pathways." Journal of Machine Learning Research. 24(240): 1–113.*
7. a, b, c, d *Kaplan J, McCandlish S, Henighan T, Brown TB, Chess B, Child R, Gray S, Radford A, Wu J, Amodei D (2020). Scaling laws for neural language models. ArXiv Preprint ArXiv:2001.08361.*
8. a, b, c *Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2019). "Language models are unsupervised multitask learners". OpenAI Blog. 1(8): 9.*
9. a, b, c *Holtzman A, Buys J, Forbes M, Choi Y (2020, April 28). "The curious case of neural text degeneration". ICLR.*

10. a, b Zhang Y. (2019). Dialogpt: Large-Scale generative pre-training for conversational response generation. ArXiv Preprint ArXiv:1911.00536.
11. a, b, c Meister C, Cotterell R (2021). Language model evaluation beyond perplexity. ArXiv Preprint ArXiv:2106.00085.
12. ^ Basu S, Choraria M, Varshney LR (2023). "Transformers are Universal Predictors." ArXiv Preprint ArXiv:2307.07843.
13. ^ Ziemann I, Matni N, Pappas GJ. (2024). State space models, emergence, and ergodicity: How many parameters are needed for stable predictions? ArXiv Preprint ArXiv:2409.13421.
14. a, b, c, d Cornfeld IP, Fomin SV, Sinai YG (2012). Ergodic theory (Vol. 245). Springer Science & Business Media.
15. ^ Ramscar M, Hendrix P, Shaoul C, Milin P, Baayen H (2014). "The Myth of Cognitive Decline: Non-Linear Dynamics of Lifelong Learning". Topics in Cognitive Science. 6(1): 5–42. doi:10.1111/TOPS.12078.
16. ^ Katok A (1995). Introduction to the Modern Theory of Dynamical Systems. Encyclopedia of Mathematics and Its Applications, 54.
17. a, b Glattfelder JB (2019). "The Semantics of Symmetry, Invariance, and Structure". Frontiers Collection. Part F1071: 65–92. doi:10.1007/978-3-030-03633-1_3/FIGURES/1.
18. a, b, c, d, e Kauffman SA (2019). A world beyond physics: the emergence and evolution of life. Oxford University Press.
19. ^ Longo G, Montévil M, Kauffman S (2012). No entailing laws, but enablement in the evolution of the biosphere. Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation, 1379–1392.
20. ^ Longo G, Montévil M (2014). Perspectives on organisms. Springer.
21. ^ Goldenfeld N, Woese C (2011). "Life is physics: evolution as a collective phenomenon far from equilibrium". Annu. Rev. Condens. Matter Phys.. 2(1): 375–399.
22. a, b LeCun Y (2022). A path towards autonomous machine intelligence (version 09). Open Review. 62(1): 1–62.
23. ^ Zador AM. (2019). A critique of pure learning and what artificial neural networks can learn from animal brains. Nature Communications, 10(1), 3770.
24. ^ Richards BA, Lillicrap TP, Beaudoin P, Bengio Y, Bogacz R, Christensen A, Clopath C, Costa RP, de Berker A, Ganguli S (2019). "A deep learning framework for neuroscience". Nature Neuroscience. 22(11): 1761–1770.

25. ^Hassabis D, Kumaran D, Summerfield C, Botvinick M (2017). "Neuroscience-inspired artificial intelligence". Neuron. 95(2): 245–258.

26. ^Wang L, Zhang X, Su H, Zhu J. (2024). A comprehensive survey of continual learning: theory, method and application. IEEE Transactions on Pattern Analysis and Machine Intelligence.

27. ^Hadsell R, Rao D, Rusu AA, Pascanu R (2020). "Embracing change: Continual learning in deep neural networks". Trends in Cognitive Sciences. 24(12): 1028–1040.

28. ^Parisi GI, Kemker R, Part JL, Kanan C, Wermter S (2019). "Continual lifelong learning with neural networks: A review". Neural Networks. 113: 54–71.

29. ^Van de Ven GM, Tolias AS (2019). "Three scenarios for continual learning". ArXiv Preprint ArXiv:1904.07734.

30. ^Krishnamurthy A, Harris K, Foster DJ, Zhang C, Slivkins A (2024). Can large language models explore in-context? ArXiv Preprint ArXiv:2403.15371.

31. a, bSteel M, Hordijk W, Kauffman SA (2020). "Dynamics of a birth–death process based on combinatorial innovation". Journal of Theoretical Biology. 491: 110187.

32. a, b, c, d, eKauffman S (2022). "Is There a Fourth Law for Non-Ergodic Systems That Do Work to Construct Their Expanding Phase Space?" Entropy. 24(10): 1383. doi:10.3390/E24101383.

33. ^Shannon CE (1948). "A mathematical theory of communication". The Bell System Technical Journal. 27(3): 379–423.

34. ^Church K, Mercer RL (1993). "Introduction to the special issue on computational linguistics using large corpora." Computational Linguistics. 19(1): 1–24.

35. a, b, c, d, e, f, g, hVaswani A (2017). "Attention is all you need". Advances in Neural Information Processing Systems.

36. ^Wang B, Min S, Deng X, Shen J, Wu Y, Zettlemoyer L, Sun H. (2022). Towards understanding chain-of-thought prompting: An empirical study of what matters. ArXiv Preprint ArXiv:2212.10001.

37. a, b, cElhage N, Nanda N, Olsson C, Henighan T, Joseph N, Mann B, Askell A, Bai Y, Chen A, Conerly T (2021). "A mathematical framework for transformer circuits". Transformer Circuits Thread. 1(1): 12.

38. ^Fan A, Lewis M, Dauphin Y (2018). "Hierarchical neural story generation". ArXiv Preprint ArXiv:1805.04833.

39. a, b, c, d, eLebowitz JL, Penrose O (1973). "Modern ergodic theory." Physics Today. 26(2): 23–29. doi:10.1063/1.3127948.

40. ^Birkhoff GD (1931). "Proof of the ergodic theorem." Proceedings of the National Academy of Sciences. 17(12): 656–660.
41. ^Neumann Jv (1932). "Proof of the quasi-ergodic hypothesis". Proceedings of the National Academy of Sciences. 18(1): 70–82.
42. ^Ruelle D (2004). Thermodynamic formalism: the mathematical structure of equilibrium statistical mechanics. Cambridge University Press.
43. a, bWalters P. (2000). An introduction to ergodic theory (Vol. 79). Springer Science & Business Media.
44. ^Kullback S (1997). Information theory and statistics. Courier Corporation.
45. a, bPrigogine I, Stengers I (2018). Order out of chaos: Man’s new dialogue with nature. Verso Books.
46. a, bMazur P (1969). "Non-ergodicity of phase functions in certain systems." Physica. 43(4): 533–545. doi:10.1016/0031-8914(69)90185-2.
47. ^Peters O (2019). "The ergodicity problem in economics". Nature Physics. 15(12): 1216–1221.
48. a, bAnderson PW (1972). "More Is Different: Broken symmetry and the nature of the hierarchical structure of science." Science. 177(4047): 393–396.
49. ^Reed M, Simon B (1980). Methods of modern mathematical physics: Functional analysis (Vol. 1). Gulf Professional Publishing.
50. ^Ainslie J, Lee-Thorp J, de Jong M, Zemlyanskiy Y, Lebrón F, Sanghai S (2023). "Gqa: Training generalized multi-query transformer models from multi-head checkpoints." ArXiv Preprint ArXiv:2305.13245.
51. ^Pang Z, Xie Z, Man Y, Wang YX (2023). "Frozen transformers in language models are effective visual encoder layers". ArXiv Preprint ArXiv:2310.12973.
52. a, b, c, dHaken H (1989). "Synergetics: an overview". Reports on Progress in Physics. 52(5): 515.
53. a, b, cHolland JH (1992). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press.
54. ^Mitchell M (2009). Complexity: A guided tour. Oxford University Press.
55. ^England JL (2015). "Dissipative adaptation in driven self-assembly". Nature Nanotechnology. 10(11): 919–923.
56. a, bCrutchfield JP (2012). "Between order and chaos". Nature Physics. 8(1): 17–24.
57. ^Cross M, Greenside H (2009). Pattern formation and dynamics in nonequilibrium systems. Cambridge University Press.

58. a, b, c, dHolland JH (2006). "Studying complex adaptive systems". Journal of Systems Science and Complexity. 19: 1–8.

59. a, bHaken H (1973). "Introduction to synergetics". Synergetics: Cooperative Phenomena in Multi-Component Systems: 9–19.

60. a, bHaken H (1993). "Synergetics: From Pattern Formation to Pattern Recognition. Some Basic Mathematical Results". Dynamical Systems: Theory And Applications: 127.

61. ^Crawford JD (1991). "Introduction to bifurcation theory." Reviews of Modern Physics. 63(4): 991.

62. ^Guckenheimer J, Holmes P (2013). Nonlinear oscillations, dynamical systems, and bifurcations of vector fields (Vol. 42). Springer Science & Business Media.

63. ^Kielhöfer H (2006). Bifurcation theory: An introduction with applications to PDEs (Vol. 156). Springer Science & Business Media.

64. ^Troger H, Steindl A (2012). Nonlinear stability and bifurcation theory: an introduction for engineers and applied scientists. Springer Science & Business Media.

65. a, bMarković D, Gros C (2014). Power laws and self-organized criticality in theory and nature. Physics Reports. 536(2): 41–74.

66. ^Sornette D (2006). Critical phenomena in natural sciences: chaos, fractals, selforganization and disorder: concepts and tools. Springer Science & Business Media.

67. ^Cross MC, Hohenberg PC (1993). "Pattern formation outside of equilibrium". Reviews of Modern Physics. 65(3): 851.

68. a, bCilliers P (2002). Complexity and postmodernism: Understanding complex systems. routledge.

69. ^Amaral LA, Buldyrev SV, Havlin S, Salinger MA, Stanley HE (1998). "Power Law Scaling for a System of Interacting Units with Complex Internal Structure." Physical Review Letters. 80(7): 1385. doi:10.1103/PhysRevLett.80.1385.

70. a, bArnold J, Holtorf F, Schäfer F, Lörch N (2024). "Phase Transitions in the Output Distribution of Large Language Models." ArXiv Preprint ArXiv:2405.17088.

71. ^Nakaishi K, Nishikawa Y, Hukushima K (2024). Critical Phase Transition in a Large Language Model. ArXiv Preprint ArXiv:2406.05335.

72. ^Bossen AM, Mauro JC (2024). "Frozen in time: A review of non-ergodic physical systems." Journal of the American Ceramic Society. 107(12): 7939–7950.

73. ^Gould SJ (1989). Wonderful Life: The Burgess Shale and the Nature of History. WW Norton & Company.

74. ^*Villani C. (2009). Optimal transport: old and new (Vol. 338). Springer.*

75. ^*Marshall WF (2011). Origins of cellular geometry. BMC Biology. 9: 1–9.*

76. a, b*Kauffman SA (1993). The origins of order: Self-organization and selection in evolution. Oxford University Press.*

77. ^*Cortês M, Kauffman SA, Liddle AR, Smolin L (2022). "The TAP equation: evaluating combinatorial innovation in biocosmology." ArXiv Preprint ArXiv:2204.14115.*

78. ^*Koppl R, Devereaux A, Herriot J, Kauffman S (2018). A simple combinatorial model of world economic history. ArXiv Preprint ArXiv:1811.04502.*

79. a, b*Ash RB, Doléans-Dade CA (2000). Probability and measure theory. Academic press.*

80. a, b*Simon HA (2012). "The architecture of complexity". In The Roots of Logistics (pp. 335–361). Springer.*

81. ^*Schlag I, Irie K, Schmidhuber J (2021). "Linear transformers are secretly fast weight programmers". International Conference on Machine Learning: 9355–9366.*

82. a, b, c, d*Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler D, Wu J, Winter C, ... Amodei D (2020). "Language Models are Few-Shot Learners." Advances in Neural Information Processing Systems. 33: 1877–1901.*

83. ^*Dao T, Fu D, Ermon S, Rudra A, Ré C (2022). "Flashattention: Fast and memory-efficient exact attention with io-awareness". Advances in Neural Information Processing Systems. 35: 16344–16359.*

84. ^*Papadimitriou CH (2003). "Computational complexity". In Encyclopedia of computer science (pp. 260–265).*

85. ^*Jelinek F (1980). "Interpolated estimation of Markov source parameters from sparse data". Proc. Workshop on Pattern Recognition in Practice, 1980.*

86. ^*Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2020). "Exploring the limits of transfer learning with a unified text-to-text transformer". Journal of Machine Learning Research. 21(140): 1–67.*

87. ^*Manning C, Schutze H (1999). Foundations of statistical natural language processing. MIT press.*

88. ^*Haussler D, Warmuth M (2018). "The probably approximately correct (PAC) and other learning models". The Mathematics of Generalization: 17–36.*

89. ^Raaijmakers JGW (1987). "Statistical Analysis of the Michaelis-Menten Equation". Biometrics. 43(4): 793. doi:10.2307/2531533.
90. ^Heinrich R, Schuster S (2012). The regulation of cellular systems. Springer Science & Business Media.
91. ^Alon U (2019). An introduction to systems biology: design principles of biological circuits. Chapman and Hall/CRC.
92. ^Klipp E, Liebermeister W, Wierling C, Kowald A (2016). Systems biology: a textbook. John Wiley & Sons.
93. ^Bootman MD, Collins TJ, Peppiatt CM, Prothero LS, MacKenzie L, De Smet P, Travers M, Tovey SC, Seo JT, Berridge MJ (2001). "Calcium signalling—an overview." Seminars in Cell & Developmental Biology. 12(1): 3–10.
94. ^Raman M, Chen W, Cobb MH (2007). "Differential regulation and properties of MAPKs". Oncogene. 26(22): 3100–3112.
95. ^Huang C-Y, Ferrell Jr JE (1996). "Ultrasensitivity in the mitogen-activated protein kinase cascade". Proceedings of the National Academy of Sciences. 93(19): 10078–10083.
96. ^Chomsky N (2014). Aspects of the Theory of Syntax (Issue 11). MIT press.
97. ^Lansing JS (2003). Complex adaptive systems. Annual Review of Anthropology. 32(1): 183–204.
98. ^Riehl E (2017). Category theory in context (Courier Dover Publications, Ed.). Google Books.
99. a, bZhang S, Roller S, Goyal N, Artetxe M, Chen M, Chen S, Dewan C, Diab M, Li X, Lin XV. (2022). Opt: Open pre-trained transformer language models. ArXiv Preprint ArXiv:2205.01068.
100. a, b, cBiderman S, Schoelkopf H, Anthony QG, Bradley H, O'Brien K, Hallahan E, Khan MA, Purohit S, Prashanth US, Raff E (2023). "Pythia: A suite for analyzing large language models across training and scaling." International Conference on Machine Learning: 2397–2430.
101. a, bHendrycks D, Burns C, Basart S, Zou A, Mazeika M, Song D, Steinhardt J (2020). "Measuring massive multitask language understanding". ArXiv Preprint ArXiv:2009.03300.
102. a, bLord FM, Novick MR (2008). Statistical theories of mental test scores. IAP.
103. ^Attanasio G, Nozza D, Hovy D, Baralis E (2022). "Entropy-based attention regularization frees unintended bias mitigation from lists." ArXiv Preprint ArXiv:2203.09192.
104. ^Raghu M, Gilmer J, Yosinski J, Sohl-Dickstein J (2017). "Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability". Advances in Neural Information Processing Systems. 30.
105. ^Jolliffe IT (2002). Principal component analysis for special types of data. Springer.

106. ^Srivastava A, Rastogi A, Rao A, Shoeb AAM, Abid A, Fisch A, Brown AR, Santoro A, Gupta A, Garriga-Alonso A (2022). "Beyond the imitation game: Quantifying and extrapolating the capabilities of language models". ArXiv Preprint ArXiv:2206.04615.

107. ^Tishby N, Zaslavsky N (2015). "Deep learning and the information bottleneck principle". 2015 IEEE Information Theory Workshop (ITW): 1–5.

108. ^Baxter J (2000). "A model of inductive bias learning." Journal of Artificial Intelligence Research. 12: 149–198.

109. ^Gal Y, Ghahramani Z (2016). "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". International Conference on Machine Learning: 1050–1059.

110. ^Vapnik VN (1999). "An overview of statistical learning theory". IEEE Transactions on Neural Networks. 10(5): 988–999.

111. ^Achille A, Soatto S (2018). "Emergence of invariance and disentanglement in deep representations." Journal of Machine Learning Research. 19(50): 1–34.

112. ^Goodfellow I (2016). Deep learning. MIT press.

113. a, bSchaeffer R, Miranda B, Koyejo S (2024). "Are emergent abilities of large language models a mirage?". Advances in Neural Information Processing Systems. 36.

114. ^Bak P (2013). How nature works: the science of self-organized criticality. Springer Science & Business Media.

115. ^Arora S, Goyal A (2023). "A theory for emergence of complex skills in language models." ArXiv Preprint ArXiv:2307.15936.

116. ^Chen H, Yang X, Zhu J, Wang W (2024). "Quantifying Emergence in Large Language Models." ArXiv Preprint ArXiv:2405.12617.

117. ^Bereska L, Gavves E (2024). "Mechanistic Interpretability for AI Safety--A Review." ArXiv Preprint ArXiv:2404.14082.

118. ^Conmy A, Mavor-Parker A, Lynch A, Heimersheim S, Garriga-Alonso A (2023). "Towards automated circuit discovery for mechanistic interpretability." Advances in Neural Information Processing Systems. 36: 16318–16352.

119. ^Liu Z, Gan E, Tegmark M (2023). Seeing is believing: Brain-inspired modular training for mechanistic interpretability. Entropy. 26(1): 41.

120. ^Rai D, Zhou Y, Feng S, Saparov A, Yao Z (2024). "A practical review of mechanistic interpretability for transformer-based language models". ArXiv Preprint ArXiv:2407.02646.

121. ^*Elhage N, Hume T, Olsson C, Nanda N, Henighan T, Johnston S, ElShowk S, Joseph N, DasSarma N, Mann B, Hernandez D, Askell A, Ndousse K, Jones A, Drain D, Chen A, Bai Y, Ganguli D, Lovitt L, ... Olah C (2022). "Softmax Linear Units". Transformer Circuits Thread.*

122. ^*Askell A, Bai Y, Chen A, Drain D, Ganguli D, Henighan T, Jones A, Joseph N, Mann B, DasSarma N (2021). "A general language assistant as a laboratory for alignment." ArXiv Preprint ArXiv:2112.00861.*

123. ^*Doshi-Velez F, Kim B (2017). "Towards a rigorous science of interpretable machine learning". ArXiv Preprint ArXiv:1702.08608.*

124. ^*Nanda N, Chan L, Lieberum T, Smith J, Steinhardt J (2023). Progress measures for grokking via mechanistic interpretability. ArXiv Preprint ArXiv:2301.05217.*

125. ^*Alabdulmohsin IM, Zhai X, Kolesnikov A, Beyer L (2024). "Getting vit in shape: Scaling laws for compute-optimal model design." Advances in Neural Information Processing Systems. 36.*

126. ^*Assran M, Duval Q, Misra I, Bojanowski P, Vincent P, Rabbat M, LeCun Y, Ballas N (2023). "Self-supervised learning from images with a joint-embedding predictive architecture." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition: 15619–15629.*

127. ^*Ha D, Schmidhuber J (2018). "World models". ArXiv Preprint ArXiv:1803.10122.*

128. ^*Matsuo Y, LeCun Y, Sahani M, Precup D, Silver D, Sugiyama M, Uchibe E, Morimoto J (2022). Deep learning, reinforcement learning, and world models. Neural Networks. 152: 267–275.*

129. a, b*Kauffman S, Levin S (1987). "Towards a general theory of adaptive walks on rugged landscapes." Journal of Theoretical Biology. 128(1): 11–45. doi:10.1016/S0022-5193(87)80029-2.*

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.