Language Models as Hierarchical Computational Projections: A Theoretical Framework with Empirical Predictions

Karol Kowalczyk

November 9, 2025

Abstract

Large language models (LLMs) exhibit strikingly regular scaling patterns linking computational resources to capability, yet existing theories fail to explain why qualitative changes in reasoning arise only beyond certain model sizes. This paper situates LLMs within a formal hierarchy of finite computational systems, extending the *Adjoint Projections on Computational Hierarchies* framework. We propose that each model functions as a finite machine $L_n = (S_n, f_n, \pi_n)$ at hierarchy level $n_L = \lfloor \alpha \log_2 P + \beta \log_2 V + \gamma \rfloor$, where P is parameter count and V is vocabulary size. Distillation and fine-tuning instantiate the adjunction's projection (P) and collapse (C) operators, governing information compression and expansion. We derive testable predictions: (1) distillation energy should scale as $E \propto k_B T \Delta H$ plus architecture-dependent overhead, (2) emergent abilities should cluster near critical levels $n \approx 33$–35, and (3) behavioral distance should predict performance degradation under compression. We outline experimental protocols to validate these predictions across model families and discuss implications for scaling law theory, computational efficiency, and the theoretical foundations of artificial intelligence.

Status: This paper presents a theoretical framework with proposed empirical validation. Experimental results are outlined as predictions to be tested.

1 Introduction

1.1 Scaling and the puzzle of emergence

Large language model development follows empirical scaling laws where performance improves according to power relationships with model size. Kaplan et al. (2020) showed that cross-entropy loss L scales approximately as $L \propto N^{-\alpha}$ where N represents parameters or training tokens, with $\alpha \approx 0.05$–0.1. Hoffmann et al. (2022) refined these relationships through compute-optimal training schedules.

However, smooth scaling curves conceal a deeper phenomenon. Wei et al. (2022) and Ganguli et al. (2022) documented that new capabilities—multi-step reasoning, mathematical problem-solving, self-consistent planning—emerge discontinuously. A 1.3B-parameter model may completely fail tasks that a 6.7B-parameter model solves reliably. These *emergent abilities* suggest structural reorganizations analogous to phase transitions in physical systems.

The fundamental question: What determines when and why these discontinuous transitions occur? Standard scaling laws offer no answer. We need a structural theory of computation accounting for how representational capacity and information flow scale with model complexity.

1.2 Hierarchical computation as a unifying principle

We adopt a hierarchical perspective building on the *Adjoint Projections on Computational Hierarchies* framework (Kowalczyk, 2025). Computation is modeled as a nested sequence of finite machines $\{M_n\}$, each operating on state spaces of size 2^n. Higher levels simulate lower ones

through embeddings, while projections compress state spaces. The adjunction $C \dashv P$ expresses duality between expansion (collapse C) and compression (projection P).

Key insight: When mapped to LLMs, these abstract operations correspond naturally to concrete practices:

- **Projection ≈ Distillation:** Compressing a large teacher model into a smaller student
- **Collapse ≈ Fine-tuning:** Expanding or enriching representations through additional training

This mapping transforms qualitative emergence into a quantitative hypothesis: emergent behavior occurs at critical points where projections between adjacent hierarchy levels become irreversibly lossy, forcing representational reorganization.

1.3 Contributions

This paper contributes:

1. **Theoretical mapping:** Explicit correspondence between LLM attributes (parameters, vocabulary) and hierarchy levels n_L, with derived level assignment formula
2. **Testable predictions:** Three quantitative predictions regarding energy costs, emergence thresholds, and performance degradation under compression
3. **Experimental protocols:** Detailed methodology for validating predictions across 20–30 models from diverse families
4. **Unification:** Integration of scaling laws, information theory, and thermodynamic cost into a coherent hierarchical framework

1.4 Limitations and scope

Limitation 1.1 (No empirical results). This paper provides a theoretical framework with testable predictions but does not present experimental validation. The predictions remain to be tested empirically.

Limitation 1.2 (Simplified model mapping). Our treatment of neural networks as finite state machines abstracts away continuous activations, gradient dynamics, and stochastic training processes. This simplification enables mathematical tractability but may miss important aspects of neural network behavior.

Limitation 1.3 (Physical claims require validation). The thermodynamic predictions (e.g., $E \propto k_B T \Delta H$) are derived from information-theoretic principles but GPU energy consumption involves many confounding factors (memory bandwidth, parallelization overhead, cooling requirements). The predicted relationship should be viewed as a theoretical baseline requiring empirical calibration.

Limitation 1.4 (Baseline comparisons needed). This framework should be compared systematically with existing neural scaling theories, including effective theories of overparameterized networks, neural tangent kernel analyses, and empirical scaling law models.

1.5 Paper organization

Section 2 reviews related work on scaling, emergence, and information theory. Section 3 presents the theoretical framework connecting adjunction theory to neural networks. Section 4 derives the level assignment formula and proposes validation methodology. Section 5 analyzes distillation as projection with thermodynamic implications. Section 6 derives scaling predictions and identifies critical transitions. Section 7 discusses limitations and future work. Section 8 concludes.

2 Related Work

2.1 Scaling laws and performance predictability

Neural network scaling laws quantify performance-resource relationships. Kaplan et al. (2020) and Hoffmann et al. (2022) showed that loss decreases predictably with compute, enabling extrapolation from small to large models. This regularity suggests underlying invariants analogous to universal scaling in statistical mechanics. However, power laws describe *continuous* improvement but fail to capture *discrete* capability jumps.

2.2 Emergent abilities and discontinuous transitions

Wei et al. (2022) documented abrupt capability transitions: arithmetic reasoning, logical inference, and complex planning appear suddenly at specific model sizes. Schaeffer et al. (2023) argued some apparent discontinuities reflect measurement artifacts, though genuine phase changes remain. Ganguli et al. (2022) hypothesized transitions correspond to representational phase changes when information per parameter exceeds critical thresholds. However, these analyses lack mathematical models connecting emergence to computability or information geometry. The hierarchical framework provides that missing structure.

2.3 Compression, distillation, and knowledge transfer

Knowledge distillation (Hinton et al., 2015) transfers knowledge from large teachers to small students by minimizing KL divergence between output distributions. This realizes projection mathematically: compressing high-dimensional representational manifolds into lower-dimensional approximations while preserving behavioral equivalence. Empirical work (Sanh et al., 2019; Jiao et al., 2020) demonstrates large efficiency gains at the cost of reduced output diversity—measurable information loss ΔH.

2.4 Information-theoretic perspectives

The information bottleneck theory (Tishby & Zaslavsky, 2015) conceptualizes learning as trading compression against relevance. Subsequent analyses (Saxe et al., 2019) refined this for layer-level information flow but not for transitions between distinct model families. Our framework extends the bottleneck concept to entire computational hierarchies with cross-level metrics.

2.5 Computational complexity and descriptive hierarchy

Theoretically, language models function as finite automata with parametric extension. Their complexity class grows with parameter count and token diversity. Tyszkiewicz (1998, 2004, 2010) analyzed analogous hierarchies in database theory using games and Kolmogorov complexity to quantify expressivity gaps. We reinterpret these results for machine learning, where expressivity gaps manifest as capability thresholds.

Gap in existing work: No prior theory connects smooth scaling laws, discontinuous emergence, compression costs, and information theory into a unified predictive framework. Our hierarchical approach provides this unification.

3 Theoretical Framework

3.1 Review: Hierarchical computational machines

The foundation comes from *Adjoint Projections on Computational Hierarchies* (Kowalczyk, 2025). Each hierarchy level corresponds to a finite machine $M_n = (S_n, f_n, \pi_n)$ with:

- State space S_n of size $|S_n| = 2^n$
- Deterministic transition function $f_n : S_n \to S_n$
- Stationary probability distribution $\pi_n : S_n \to [0, 1]$

Information capacity is $I_n = n$ bits. Levels connect via:

- **Embeddings** $\sigma_{i\to j} : S_i \hookrightarrow S_j$ (injective, structure-preserving)
- **Projections** $P_{j\to i} : S_j \twoheadrightarrow S_i$ (surjective, entropy-minimizing)
- **Collapses** $C_{i\to j} : S_i \hookrightarrow S_j$ (injective, error-minimizing, satisfying $P \circ C = \mathrm{id}$)

The pair (C, P) forms an adjunction $C \dashv P$ with:

- Unit $\eta : \mathrm{id} \Rightarrow C \circ P$
- Counit $\varepsilon : P \circ C \Rightarrow \mathrm{id}$
- Triangle identities: $(\varepsilon P) \circ (P\eta) = \mathrm{id}_P$ and $(C\varepsilon) \circ (\eta C) = \mathrm{id}_C$

Behavioral distance $\mathrm{Beh}(i, j)$ quantifies computational divergence between levels via normalized Hamming disagreement on common embedded domains. The cross-level metric $d(M_i, M_j) = |2^{-i} - 2^{-j}| + 2^{-\min(i,j)} \cdot \mathrm{Beh}(i, j)$ defines a metric space with completion T_c (the *computational continuum*).

3.2 Mapping language models to hierarchy levels

An LLM maps onto this framework as follows:

States (S_n)**:** The state space comprises:

- Hidden representations $h \in \mathbb{R}^d$ at each layer
- Token embeddings $e \in \mathbb{R}^{d_{\mathrm{embed}}}$
- Attention patterns and intermediate activations

Effective state space size scales with parameter count P and embedding dimension d: $|S_n| \propto 2^d \cdot P$ (heuristically, since P parameters define a P-dimensional weight space over quantized values).

Transitions (f_n)**:** Layer-wise transformations implement f_n:

$$f_n(h) = \mathrm{LayerNorm}(h + \mathrm{MHA}(h) + \mathrm{FFN}(h))$$

where MHA is multi-head attention and FFN is feed-forward network. Each layer maps hidden states deterministically (during inference with fixed weights).

Distributions (π_n)**:** Output softmax defines π_n:

$$\pi_n(\mathrm{token} \mid \mathrm{context}) = \mathrm{softmax}(W_{\mathrm{out}} \cdot h_{\mathrm{final}})$$

where h_{final} is the final hidden state and W_{out} projects to vocabulary.

Capacity correspondence: Model capacity corresponds to $\log_2 |S_n|$, which should scale with:

- $\log_2 P$: Parameter count determines weight space dimension
- $\log_2 V$: Vocabulary size determines output space dimension

- Architecture factors: Depth, width, attention heads contribute multiplicatively

Thus each LLM instantiates M_n at some hierarchy level n_L.

Remark 3.1 (On the finite state machine abstraction). As noted in Limitation 1.2, this mapping abstracts continuous neural network dynamics into discrete state transitions. While gradient descent operates in continuous parameter space and activations are real-valued, the effective computational behavior—especially at inference time—can be approximated by finite precision arithmetic. Modern GPUs use FP16 or INT8 quantization, making the finite state abstraction more realistic than it initially appears. However, the mapping remains approximate and experimental validation is essential.

3.3 Adjoint operations in practice

Projection (Distillation): Teacher model M_j distills to student M_i ($i < j$) by minimizing KL divergence:

$$L_{\text{distill}} = \mathbb{E}_x[\text{KL}(P_{\text{teacher}}(\cdot|x) \| P_{\text{student}}(\cdot|x))]$$

This compresses high-entropy teacher outputs into lower-entropy student approximations. Information loss is:

$$\Delta H = H(P_{\text{teacher}}) - H(P_{\text{student}})$$

Collapse (Fine-tuning): Starting from pre-trained model M_i, fine-tuning on domain-specific data enriches representations toward effective level M_j ($j > i$). Formally:

$$L_{\text{finetune}} = \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{target}}}[\text{CE}(y, M_i(x))]$$

where $\mathcal{D}_{\text{target}}$ is the target domain dataset and CE is cross-entropy loss.

Adjunction property: If distillation from M_j to M_i yields $\hat{M}_i$ and subsequent fine-tuning recovers $\hat{M}_j$, then:

$$P_{j\to i}(C_{i\to j}(\hat{M}_i)) \approx \hat{M}_i \quad \text{(near-identity under projection-collapse)}$$

This is the discrete analog of the counit $\varepsilon : P \circ C \Rightarrow \text{id}$.

Remark 3.2 (Approximate vs. exact adjunction). In practice, the adjunction holds only approximately due to optimization error, data distribution shift, and architectural constraints. The theoretical framework provides an idealized target; empirical validation should quantify deviation from exact adjunction.

4 Level Assignment and Validation

4.1 Deriving the level formula

Proposition 4.1 (Level assignment formula)**.** *The hierarchy level n_L of an LLM with P parameters and vocabulary size V is:*

$$n_L = \lfloor \alpha \log_2 P + \beta \log_2 V + \gamma \rfloor$$

where α, β, γ are empirically determined constants.

Derivation. Effective state space size $|S_n|$ scales with:

- Parameter count: Each parameter can take $\approx 2^b$ values (for b-bit quantization), giving P^{2^b} configurations. In log-space: $\log_2 |S_n| \approx b \cdot \log_2 P$.
- Vocabulary: Output space is V^{context} for contexts of bounded length, contributing $\log_2 V$ to capacity.

- Architectural factors: Depth L, width W, attention heads H contribute multiplicatively, absorbed into γ.

Setting $\alpha \approx b$ (quantization bits), $\beta \approx 1$ (first-order vocab contribution), and γ as an architectural offset yields the formula. Empirical fitting to real models determines exact values. □

Remark 4.2 (Expected coefficient values). Based on typical neural network architectures:

- $\alpha \approx 0.5$–1.0 (reflecting effective parameter precision)
- $\beta \approx 0.3$–0.7 (vocabulary contributes less than parameters)
- $\gamma \approx 10$–20 (baseline architectural complexity)

These should be fitted empirically using benchmark performance data.

4.2 Validation methodology

To validate the level assignment:

1. **Compute** n_L for 20–30 models across families (GPT, LLaMA, Falcon, Pythia, MPT)
2. **Measure performance** on benchmark tasks (GSM8K, Big-Bench Hard, HumanEval, GLUE)
3. **Fit coefficients** (α, β, γ) by regression: performance $\sim f(n_L)$
4. **Predict** performance for held-out models and compare to actual measurements
5. **Assess goodness of fit**: $R^2 > 0.85$ would indicate strong predictive power

5 Distillation as Projection: Thermodynamic Analysis

5.1 Information-theoretic energy cost

Prediction 5.1 (Distillation energy scaling)**.** The energy required to distill teacher M_j to student M_i should scale as:

$$E_{\text{distill}} = k_B T \cdot \Delta H + E_{\text{arch}}$$

where:

- $k_B T$ is thermal energy ($T \approx 300$K for GPU operation)
- $\Delta H = H(P_{\text{teacher}}) - H(P_{\text{student}})$ is information loss in bits
- E_{arch} is architecture-dependent computational overhead

Remark 5.2 (Landauer's principle connection). This prediction extends Landauer's principle (minimum energy $k_B T \ln 2$ per bit erased) to neural network compression. Each bit of entropy lost in distillation requires thermodynamic work. However, GPU energy consumption includes many non-thermodynamic factors (see Limitation 1.3), so the relationship is expected to hold only as a baseline with substantial overhead.

5.2 Experimental protocol

To test Prediction 5.1:

1. **Select teacher-student pairs** spanning wide ΔH range (0.5–5 bits)
2. **Measure distillation energy** using GPU power monitoring (detailed in Appendix A.3 of the original manuscript)
3. **Quantify** ΔH via entropy difference on validation set
4. **Fit linear model**: $E = a \cdot \Delta H + b$ and compare a to $k_B T \ln 2 \approx 4.3 \times 10^{-21}$ J
5. **Interpret discrepancy**: Large E_{arch} indicates overhead dominates; proportionality still validates information-theoretic scaling

Remark 5.3 (Confounding factors). GPU energy consumption depends on:

- Memory bandwidth and cache efficiency
- Parallelization overhead and synchronization
- Floating-point operation throughput
- Cooling and power regulation circuits

These factors can dominate the thermodynamic minimum. The prediction should be interpreted as: *after controlling for architecture and hardware, energy should scale proportionally with information loss.* This requires careful experimental design with controlled hardware and multiple model families.

6 Scaling Predictions and Critical Transitions

6.1 Emergence thresholds

Prediction 6.1 (Critical transition levels)**.** Emergent abilities should appear near hierarchy levels $n \approx 33$–35, corresponding to models with $\approx$ 6B–13B parameters (assuming typical vocabulary $V \approx 50$K and $\alpha \approx 0.8, \beta \approx 0.5, \gamma \approx 15$).

Rationale: If capabilities require representational capacity exceeding $2^{33} \approx 8.6 \times 10^9$ effective states, smaller models cannot implement the necessary computational circuits. The discontinuity arises from discrete level structure.

Remark 6.2 (Falsifiability). Prediction 6.1 is highly falsifiable. If emergent abilities appear uniformly across all model sizes or at substantially different thresholds ($n < 30$ or $n > 38$), the hierarchical framework would require revision. Schaeffer et al. (2023) suggest some emergence may be measurement artifacts; distinguishing genuine phase transitions from smooth capability growth with nonlinear metrics is crucial.

6.2 Compression-performance tradeoff

Prediction 6.3 (Behavioral distance and performance degradation)**.** Performance degradation under distillation should correlate with behavioral distance:

$$\Delta\text{Acc} \propto \text{Beh}(n_{\text{teacher}}, n_{\text{student}}) + \Delta H$$

where ΔAcc is accuracy loss on benchmark tasks.

Validation: Measure performance before/after distillation across model pairs, compute behavioral distance via output distribution divergence (KL divergence normalized by entropy), and fit regression model.

6.3 Optimal distillation ratios

The framework predicts diminishing returns when compressing beyond certain ratios:

Corollary 6.4 (Distillation efficiency bound)**.** *For teacher at level n_j and student at level n_i, distillation efficiency $\eta = Acc_{student}/Acc_{teacher}$ should satisfy:*

$$\eta \geq \exp(-C \cdot (n_j - n_i))$$

for some constant $C > 0$. This gives exponential performance decay with level gap.

7 Discussion

7.1 Theoretical limitations

7.1.1 Finite state abstraction

As acknowledged in Limitation 1.2, treating neural networks as finite state machines ignores:

- **Continuous activations:** Real-valued hidden states have uncountably infinite precision (before quantization)
- **Gradient dynamics:** Training involves continuous optimization in parameter space
- **Stochasticity:** Sampling introduces randomness not captured by deterministic transitions

However, at inference time with fixed weights and finite precision arithmetic (FP16, INT8), the effective behavior is approximately finite-state. The degree of approximation quality is an empirical question.

7.1.2 Thermodynamic predictions

As noted in Limitation 1.3 and Remark 5.3, GPU energy consumption has many confounding factors. The $E \propto k_B T \Delta H$ relationship should be viewed as a theoretical baseline. Empirical validation may reveal:

- Large architecture-dependent overhead $E_{\text{arch}} \gg k_B T \Delta H$
- Non-trivial dependence on batch size, sequence length, and parallelization strategy
- Hardware-specific effects (memory hierarchy, interconnect bandwidth)

The prediction remains valuable if energy scales *proportionally* with ΔH after controlling for these factors, even if the absolute magnitude differs from theoretical minimum.

7.2 Relation to existing theories

7.2.1 Neural scaling laws

Our framework complements rather than replaces empirical scaling laws. Kaplan et al. (2020) and Hoffmann et al. (2022) provide phenomenological power-law fits. We offer a structural explanation: hierarchy levels impose discrete capacity thresholds, and the smooth average conceals stepwise transitions.

7.2.2 Neural tangent kernel theory

NTK theory analyzes overparameterized networks in the infinite-width limit where training becomes linear. Our finite-hierarchy framework applies to practical-sized models where discrete resource constraints matter. The two perspectives address different regimes.

7.2.3 Information bottleneck

Tishby & Zaslavsky (2015) analyze layer-level compression-relevance tradeoffs. We extend this to model-level hierarchies across different architectures. The adjunction structure provides mathematical formalism for compression/expansion operations.

7.3 Future work

7.3.1 Empirical validation

Priority experiments:

1. Fit level assignment formula to 30+ models and validate on held-out test set
2. Measure distillation energy for 10+ teacher-student pairs across different ΔH ranges
3. Map emergence thresholds for specific capabilities (arithmetic, reasoning, code generation)
4. Quantify behavioral distance via output distribution divergence and correlate with performance

7.3.2 Theoretical extensions

- **Continuous limit:** Replace discrete hierarchy with differential equations governing level transitions
- **Multi-modal models:** Extend framework to vision-language models with cross-modal projections
- **Training dynamics:** Incorporate gradient descent as trajectory through hierarchy levels
- **Formal expressivity bounds:** Prove rigorous capacity limits at each hierarchy level

7.3.3 Practical applications

- **Compression algorithms:** Design distillation protocols minimizing $\mathrm{Beh}(n_T, n_S)$
- **Architecture search:** Optimize network design for target hierarchy level
- **Capability prediction:** Estimate minimum model size for specific tasks from information-theoretic bounds

8 Conclusion

We have presented a theoretical framework situating large language models within a formal computational hierarchy. The key insights:

1. **Structural mapping:** LLMs instantiate finite machines at hierarchy levels determined by parameter count and vocabulary size
2. **Adjoint operations:** Distillation and fine-tuning realize projection/collapse operators with information-theoretic constraints
3. **Testable predictions:** Energy scaling, emergence thresholds, and compression-performance tradeoffs provide empirical falsification criteria
4. **Theoretical unification:** The framework connects scaling laws, information theory, and thermodynamics through categorical structure

The framework's value lies not in immediate empirical confirmation but in providing *precise quantitative predictions* that can be systematically tested. The specific threshold $n \approx 33$–35 for emergence, the energy-entropy relationship $E \propto k_B T \Delta H$, and the behavioral distance-performance correlation are all falsifiable hypotheses.

Limitations are substantial: the finite-state abstraction is approximate, thermodynamic predictions require careful interpretation, and empirical validation is entirely absent. However, these limitations are clearly stated, and the framework provides sufficient mathematical structure to guide experimental design.

If validated, this approach could transform our understanding of model scaling from phenomenological observation to principled theory grounded in computation and information. If falsified, the systematic exploration will reveal which aspects of neural networks defy hierarchical description—equally valuable for scientific progress.

References

[1] Ganguli, D., et al. (2022). Predictability and surprise in large generative models. In *Proceedings of FAccT*.

[2] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

[3] Hoffmann, J., et al. (2022). Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

[4] Jiao, X., et al. (2020). TinyBERT: Distilling BERT for natural language understanding. In *Proceedings of EMNLP*.

[5] Kaplan, J., et al. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

[6] Kowalczyk, K. (2025). Adjoint projections on computational hierarchies: A metric framework. *Manuscript in preparation*.

[7] Sanh, V., et al. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

[8] Saxe, A. M., et al. (2019). On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12), 124020.

[9] Schaeffer, R., Miranda, B., & Koyejo, S. (2023). Are emergent abilities of large language models a mirage? *arXiv preprint arXiv:2304.15004*.

[10] Tishby, N., & Zaslavsky, N. (2015). Deep learning and the information bottleneck principle. In *Proceedings of ITW*.

[11] Tyszkiewicz, J., & Vianu, V. (1998). Queries and computation on the web. In *Proceedings of ICDT*, 275–289.

[12] Tyszkiewicz, J. (2004). On asymptotic probabilities of monadic second order properties. In *Proceedings of ICALP*, 887–899.

[13] Tyszkiewicz, J. (2010). Kolmogorov complexity and expressive power. *Information and Computation*, 208(7), 729–743.

[14] Wei, J., et al. (2022). Emergent abilities of large language models. *Transactions on Machine Learning Research*.