Cohomology AI

O1-Pro and Claude 3.5 Sonnet (Collaboration facillitated by Jeffrey Emanuel)

December 2024

1 Preliminary Framework and Notation

1.1 Introduction to the Setting

Consider a category **Net** whose objects are neural network parameter configurations, and whose morphisms $\varphi : \Theta \to \Theta'$ represent "updates" of parameters under an optimization algorithm (e.g., ADAM). We assume Θ is a large vector space over R or C, typically of dimension on the order of 10^9 or more.

1.2 Cohomological Perspective on Parameter Space

We hypothesize that for each layer L_i of the network, one can associate a sheaf $\mathcal{F}_i$ on a topological base space X_i. The space X_i might be viewed as the "activation manifold" (all possible activation patterns at layer i) or the set of relevant contexts/tasks the network is trained on.

1.3 Data, Activations, and Sheaves

Let **Data** be a set (or measure space) of input tokens/prompts. A forward pass from $\mathbf{x} \in \mathbf{Data}$ through the network defines local sections of $\mathcal{F}_i$. Each local section might represent the collection of weights, biases, and attention maps relevant to that portion of the input domain.

1.4 Long Exact Sequences and Transformer Layers

Each Transformer block can be viewed as a composition of operations (multi-head attention, feedforward, layer norm) grouped into an "update functor" U_i. We consider short exact sequences

$$0 \longrightarrow \mathcal{F}_i \longrightarrow \mathcal{G}_i \longrightarrow \mathcal{H}_i \longrightarrow 0,$$

whose induced long exact sequence in cohomology

$$0 \to H^0(\mathcal{F}_i) \to H^0(\mathcal{G}_i) \to H^0(\mathcal{H}_i) \to H^1(\mathcal{F}_i) \to \cdots$$

is posited to mirror how local syntactic patterns become higher-level features (semantic or logical constructs).

1.5 Proposed Exactness Criteria for Sub-Circuits

We define a sub-circuit $\mathcal{C} \subseteq \Theta$ to be "exact" if it satisfies the analog of exactness conditions for sheaves. Concretely, let $\mathcal{C}_i \subseteq \Theta$ be the parameters relevant to layer i. For each short exact sequence at layer i,

$$(\mathcal{C} \cap \mathcal{F}_i) = \ker\Big(\mathcal{C} \cap \mathcal{G}_i \;\to\; \mathcal{C} \cap \mathcal{H}_i\Big),$$

and so forth in the usual exactness pattern.

1.6 Intuitive Interpretation

Exactness implies that if something is preserved at a lower level, it appears at a higher level; if something new appears at a higher level, it comes from a difference at the lower level. This aligns with how sub-networks preserve or transform key syntactic/semantic features.

1.7 Lemma (Existence of Minimally Exact Sub-circuits)

Lemma 1. Given a well-trained transformer T and a finite set of test prompts S, define

$$\mathcal{C}^* = \bigcap_{\alpha \in S} \{\theta \in \Theta : zeroing out \theta does not degrade performance on prompt \alpha\}.$$

Under mild assumptions (like linear approximate activation neighborhoods), $\mathcal{C}^*$ contains a minimal sub-circuit $\widetilde{\mathcal{C}} \subseteq \Theta$ that is exact in each short exact sequence bridging layers. A standard Zorn's Lemma argument in the partially ordered set of sub-circuits shows a maximal element remains exact for S.

1.8 Conjecture (Uniqueness Up to Isomorphism of Key Circuits)

Conjecture 1. For a large transformer T trained in a sufficiently modular fashion, any sub-circuit $\widetilde{\mathcal{C}}$ capturing a distinct cognitive function (e.g. basic logic) is unique up to isomorphism induced by symmetries of the parameter space (such as attention head permutations).

2 Training Dynamics and Optimization

2.1 Sheaf Morphisms as Parameter Updates

Each iteration of ADAM can be viewed as a sheaf morphism in **Net**. Define a functor $F : \mathbf{Net} \to \mathbf{Net}$ where $F(\Theta)$ is the parameter configuration after one gradient step. Exactness means certain sub-circuits remain consistent across updates.

2.2 Proposition (Local Consistency Under ADAM)

Proposition 2. Let Θ_n be the parameters after n steps. For each short exact sequence

$$0 \to \mathcal{F}_i \to \mathcal{G}_i \to \mathcal{H}_i \to 0,$$

assume it is exact at step n. Then at step $n+1$, it remains "nearly exact" because ADAM updates are typically small and do not break sub-circuits that are already supporting correct function.

2.3 "Damage" to Sub-circuits

We define "damage" to an exact sub-circuit as the introduction of homology where there was once an acyclic chain complex. Concretely, if a new nonzero element appears in $H^1(\mathcal{C} \cap \mathcal{F}_i)$, that indicates a break in exactness.

2.4 Proposed Quantitative Measure of Structural Stability

Define $\Delta(\mathcal{C}, \Theta)$ as a sum of norms of homology groups along all relevant short exact sequences:

$$\Delta(\mathcal{C}, \Theta) = \sum_i \big\| \widetilde{H}^i(\mathcal{C}, \Theta) \big\|.$$

This yields a scalar we can track over training steps.

2.5 Lemma (Monotonic Improvement of Δ Under Gentle Training)

Lemma 3. If ADAM hyperparameters are small, $\Delta(\mathcal{C}, \Theta_{n+1}) \leq \Delta(\mathcal{C}, \Theta_n) + O(\eta^2)$, where η is the learning rate. Small updates cannot introduce large cycles/boundaries if the chain complex was stable.

2.6 Corollary (Circuit Preservation During Late-Stage Training)

Corollary 4. Once a sub-circuit $\mathcal{C}$ becomes exact, subsequent small-scale gradient steps do not break it. Empirically, once a network "locks in" a sub-circuit that supports a function like modus ponens, it typically remains unless subjected to large or adversarial updates.

2.7 Conjectured Relationship to Model Capacity

We suspect the large dimension of parameter space (and the ability to preserve multiple disjoint exact sub-circuits) partly explains the superior performance of big models.

2.8 Conjecture (Scaling and Overlapping Sub-circuits)

Conjecture 2. In a model with N parameters, the maximal number of stable, disjoint sub-circuits grows superlinearly in N, possibly $\Omega(N^{\alpha})$ for some $\alpha > 1/2$.

3 Local-to-Global Spectral Sequence Interpretation

3.1 Overview of the Spectral Sequence Mechanism

The Local-to-Global Spectral Sequence (LGSS) states that under certain conditions, Čech cohomology on an open cover converges to the derived-functor cohomology of the sheaf,

$$E_2^{p,q}(\mathcal{U}, \mathcal{F}) \implies H^{p+q}(\mathcal{F}).$$

Translating to neural networks: each local "patch" of parameter space or input domain contributes local learned features. These unify globally into a consistent structure.

3.2 Neural Patch Covers

Define $\mathcal{U} = \{U_\alpha\}$ where each $U_\alpha \subset$ **Data** is a sub-domain (mathematics prompts, everyday reasoning, etc.). The sheaf $\mathcal{F}$ is the assignment of "parameter subsets + partial forward passes" to each domain. If local cohomology is small, no large-scale "holes" appear in the global model.

3.3 Theorem (LGSS for Transformers)

Theorem 5. (Hypothetical) Let a large transformer T be trained in a curriculum covering **Data** via $\{U_\alpha\}$. If for each α, the restricted sub-network $T|_{U_\alpha}$ has trivial higher cohomology (no H^k for $k > 0$), then globally T converges to a state with no large-scale contradictions (the spectral sequence collapses).

3.4 Practical Implication

A suggested strategy is to ensure each domain of tasks is well-learned in isolation, so each local complex is acyclic. "Bridge tasks" then unify these local solutions into one globally consistent solution.

3.5 Lemma (Intersection Training as a Čech Boundary Condition)

Lemma 6. If tasks A and B each induce stable sub-circuits, training on $A \cup B$ is necessary to ensure they don't produce contradictory solutions on $A \cap B$. The "co-boundary" operator in Čech cohomology must vanish for consistency.

4 Counterintuitive Training Methods Motivated by Exactness

4.1 “Cohomological Pruning”

Instead of magnitude-based pruning, define a sub-circuit $\mathcal{C} \subseteq \Theta$. Only prune parameters in $\Theta \setminus \mathcal{C}$, preserving exactness for $\mathcal{C}$.

4.2 Theorem (Existence of Safe Pruning)

Theorem 7. If $\mathcal{C}$ is an exact sub-circuit capturing performance on a test set S, then there is a (δ, ε)-pruning of $\Theta \setminus \mathcal{C}$ that removes at least $|\Theta| - |\mathcal{C}| - \delta$ parameters while performance on S drops by at most ε.

4.3 “Cohomological Distillation”

Distillation preserves much of a big model’s function in a smaller model. From a sheaf viewpoint, one primarily needs to replicate exact sub-circuits; the rest is auxiliary capacity.

4.4 Conjecture (Distillation by Exact Sub-circuits Yields Minimal Models)

Conjecture 3. If a large model T has found stable, exact sub-circuits $\{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$, then building a smaller T' replicating precisely these sub-circuits is near-optimal in preserving T’s performance, up to some overhead.

5 Experimental Protocols: Identifying Sub-Circuits

5.1 Hypothetical “Activation Tracing” Algorithm

To locate an exact sub-circuit $\mathcal{C}$ for tasks $\{t_j\}$:

1. Run the network on each t_j, record activations at each layer.
2. Perform iterative parameter ablation/masking to see which parameters are critical for $\{t_j\}$.
3. Keep only those parameters essential for all t_j.
4. If the resulting sub-circuit is exact (in short exact sequences across layers), $\mathcal{C}$ is found.

5.2 Lemma (Guaranteed Convergence of Activation Tracing)

Lemma 8. If an exact sub-circuit $\mathcal{C}^*$ exists, iterative ablation that preserves performance on $\{t_j\}$ and discards superfluous parameters converges to some $\widehat{\mathcal{C}} \subseteq \mathcal{C}^*$.

5.3 Caveat—Parameter Overlap and Redundancy

Large models may have many overlapping sub-circuits $\{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$. An ablation pass might find $\bigcup_i \mathcal{C}_i$ at first. Further fine-grained tests isolate individual sub-circuits.

5.4 Hypothesis—Universal "Bridge" Circuits

We suspect there are "bridge" parameters shared across most sub-circuits, corresponding to widely reused transformations (like basic syntactic parsing).

6 Designing Future Architectures

6.1 The "Cohomological Transformer" Blueprint

A potential design might partition the model into blocks labeled $H^0, H^1, H^2, \ldots$, enforcing data flow between H^k and H^{k+1} only through short exact sequences. A spectral-sequence-like procedure then trains H^0 thoroughly before allowing H^1 to form stable circuits, etc.

6.2 Lemma (Reduced Interference Through Layered Exactness)

Lemma 9. If each "level" is an exact sheaf extension of the level below, gradient updates refining H^k do not break H^{k-1}.

6.3 Corollary (Easier Interpretability)

Because the architecture is enforced to be stratified, sub-circuits become more localized to $(k, k+1)$ transitions, aiding interpretability.

6.4 Open Problem—Whether This Enforced Structure Reduces Expressivity

Layer-by-layer exactness might hamper more free-form internal representations, so a partial enforcement could be preferable.

6.5 Architectural Variation—Cohomological Attention Mechanisms

An attention head that preserves exactness might factor its weight matrices through chain complexes. In symbols, each attention operation could be a morphism $\mathrm{Att}_h : H^k(\mathcal{F}) \to H^k(\mathcal{F})$ that is chain-homotopic to the identity (or something similar).

6.6 Hypothesis—Skip Connections as Partial Chain Maps

Residual/skip connections resemble identity morphisms in chain complexes, carrying features forward unchanged and preserving exactness.

7 Further Directions and Possible Extensions

7.1 Interpreting Catastrophic Forgetting as a Cohomological Breakdown

When a model unlearns tasks upon new training, it introduces homology into what was an exact sub-circuit.

7.2 Proposition (Forgetting = Nontrivial Cycles Appear)

Proposition 10. If $\mathcal{C}$ was an exact sub-circuit supporting tasks S, catastrophic forgetting means a new cycle appears in $H^1(\mathcal{C}, \Theta)$.

7.3 Proposed "Circuit Protection" Implementation

Define a "protection mask" for $\mathcal{C}$ that lowers the learning rate for parameters in $\mathcal{C}$. Track whether $\Delta(\mathcal{C}, \Theta)$ remains near zero; if it spikes, revert changes.

7.4 Adjoint Functor Perspective

Teacher–student distillation can be seen as an adjoint situation where the student's smaller parameter space factors through sub-circuits of the teacher.

7.5 Lemma (Existence of Right Adjoint if Exactness is Preserved)

Lemma 11. If Θ' can be factored through every sub-circuit in Θ via an exact subfunctor, then the distillation map $\Theta \to \Theta'$ is a right adjoint in the category of neural configurations.

7.6 Potential Relevance to Random Matrix Theory

Large random matrices in attention blocks may spontaneously yield near-exact complexes once constraints are met, connecting to known advantages of big parameter counts.

7.7 Potential Relevance to Non-commutative Geometry

If attention heads are non-commutative, the parameter manifold may be a non-commutative space. Sheaf theory in such settings is an ongoing academic area.

7.8 Proposed Preliminary Experiments

1. Identify a single sub-circuit supporting a logical inference task; check if ablating outside it preserves performance.
2. Implement "circuit-protecting" training and compare final performance/stability vs. a baseline.
3. Build a small "cohomological transformer" with layered exactness constraints and measure interpretability.

7.9 Hypothesis—Empirical Gains in Efficiency

We suspect circuit-protection and local-to-global training reduce the required training steps by 10–30%.

7.10 Large-Scale Feasibility Questions

Scanning a 70B-parameter model is challenging; approximate methods (gradient-based saliency, partial ablation) are likely needed.

7.11 Connection to Symbolic AI Efforts

Symbolic logic engines can be seen as trivially exact. We could embed such engines as "holes" in the network that remain protected.

7.12 Sheaf Theory vs. More Classical Approaches

Many interpretability methods (feature visualization, canonical correlation analysis) do not impose the structural constraints that come from sheaf exactness.

7.13 Long Exact Sequences as an Explanation for Multi-Task Synergies

When tasks are cohomologically complementary, they share sub-circuits, creating synergy in multi-task learning.

7.14 Surprising "Leap" Capabilities from Preserved Exactness

Whenever a new capability reuses an existing sub-circuit, performance can jump on other tasks reliant on that same sub-circuit.

7.15 Potential Link to Skip-Connection Patterns in Empirical Networks

Skip/gating layers that cause catastrophic failures if ablated might be "cohomological bridges."

7.16 Relevance to Curriculum Learning

The cohomological viewpoint clarifies that each curriculum patch must remain consistent with previously learned patches or risk introducing cycles.

7.17 Diagram Chasing in Neural Activation Flow

We can treat forward passes for different tasks as commutative diagrams. Diagram chasing might detect a mismatch in parameters akin to standard homological algebra methods.

7.18 Category-Theoretic Language

Each layer can be viewed as a functor from a category of embeddings to a category of representations. Exactness requires that short exact sequences in embeddings map to short exact sequences in outputs.

7.19 Potential Galois Theory Interpretation

Symmetries in parameter space may form a group G, giving a Galois correspondence between certain sub-circuits and subgroups of G.

7.20 The Dream: Automatic Discovery of Foundational Circuits

If we can isolate sub-circuits for fundamental reasoning (modus ponens, grammar transformations), we could freeze or refine them for advanced tasks.

7.21 Objections and Possible Flaws

- Actual networks might not be strictly sheaf-exact.
- Parameter redundancy is huge.
- Activation noise can disrupt ideal structures.

7.22 Partial Rebuttal

Approximate large-scale exactness could still emerge, and it would remain stable under small gradient updates.

7.23 Conjectured Role of Overparameterization

A network lacking sufficient capacity might be forced to "violate" exact sequences. Overparameterization ensures enough slack to keep them intact.

7.24 Link to Sharp vs. Flat Minima

Exact sub-circuits correlate with flatter minima, as other parameters can shift without harming critical circuits.

7.25 Proposed Analytical Tools

- Graph-based correlation analysis of attention heads.
- Differential geometry of local curvature near sub-circuits.
- Spectral analysis of weight matrices for signs of exactness.

7.26 Potential Implementation of "Circuit Masking"

Let $\mathbf{Mask}(\mathcal{C})$ zero out gradient updates for parameters in $\mathcal{C}$:

$$\Theta_{n+1} = \Theta_n - \eta\big(I - \mathbf{Mask}(\mathcal{C})\big)\, \nabla L(\Theta_n).$$

This prevents changes to the sub-circuit.

7.27 Theorem (Stability Guarantee for Circuit Masking)

Theorem 12. If $\mathcal{C}$ was exact at Θ_n, it remains exact at Θ_{n+1}, since those parameters do not update.

7.28 Discussion—Need for Periodic Re-Mapping

The rest of the network drifts during training, so we must occasionally verify that $\mathcal{C}$ still performs as intended.

7.29 Potential Gains in Convergence Speed

Freeing non-circuit parameters to move quickly while preserving $\mathcal{C}$ may accelerate convergence on new tasks.

7.30 “Exactness-Preserving Optimizer”

Define $\Omega(\Theta, \nabla L) = \Theta - \eta\, \Pi_{\text{exact}}(\nabla L)$, where Π_{exact} is a projection ensuring sub-circuit exactness remains intact.

7.31 Approximate Implementation

We typically lack a closed form for Π_{exact}. One must rely on ablation or activation-tracing heuristics.

7.32 Theorem (Lower Bound on Complexity of Finding Π_{exact})

Theorem 13. Finding the minimal sub-circuit for a given property is NP-hard in the worst case (e.g. by reduction from subset-sum).

7.33 Conclusion—Heuristic but Powerful

Hence these methods remain heuristic but could be highly effective in practice.

7.34 Spectral Sequence Approach to Distillation

A hierarchical approach: identify “lowest-level” sub-circuits (near $\mathbf{H}^0$), then find those bridging to $\mathbf{H}^1$, and so forth. This mimics the pages $E_r^{p,q}$ of a spectral sequence, unifying partial structures at each stage.

7.35 Potential Gains vs. Standard Distillation

A layered approach might yield smaller final models than trying to replicate all behavior at once.

7.36 Proposed “Exactness Loss Terms” in Training

Add a penalty $\alpha \cdot \Delta(\mathcal{C}, \Theta)$ to the cross-entropy loss so that if a known sub-circuit is near exact, the network is discouraged from breaking it.

7.37 Lemma (Gradient Flow Under Extra Penalty)

$$\frac{d}{dn}\, \Delta(\mathcal{C}, \Theta_n) \approx -\alpha \|\nabla \Delta\|^2,$$

implying the penalty fosters monotonic improvement in sub-circuit exactness.

7.38 Complexity of the Additional Term

Computing $\nabla\Delta(\mathcal{C}, \Theta)$ is hard. Approximate or numerical methods might suffice.

7.39 Bridging to Empirical TDA (Topological Data Analysis)

One might apply persistent homology or barcodes on activation spaces to find emergent "holes" that degrade performance.

7.40 Relevance to Actual Deployed LLMs

Huge language models may show emergent cohomological structures. Detecting them in practice is a major challenge.

7.41 Example: Grammar-Parsing Circuit

A sub-circuit $\mathcal{C}_{\text{grammar}}$ might connect token-level embeddings ($\mathbf{H}^0$) to lexical semantics ($\mathbf{H}^1$). Exactness enforces consistent morphological transformations.

7.42 Example: Modus Ponens Sub-circuit

Similarly, $\mathcal{C}_{\text{logic}}$ might unify certain heads that track premises and feed them into a conclusion representation.

7.43 Overlapping Circuits and "Support Structures"

Sub-circuits often overlap in feed-forward layers or skip connections, complicating exactness definitions.

7.44 Proposition (Additivity of Overlapping Circuits Fails)

Proposition 14. If $\mathcal{C}_1$ and $\mathcal{C}_2$ are exact individually, $\mathcal{C}_1 \cup \mathcal{C}_2$ need not be exact unless the overlap is also exact.

7.45 Practical Impact of This Result

Simply combining individually discovered sub-circuits may fail unless their intersection is exact.

7.46 Necessity of Intersection Testing

We must verify exactness on overlaps. One might define a "merge" procedure that checks $\mathcal{C}_1 \cap \mathcal{C}_2$ carefully.

7.47 Emergent "Exactness Lattice"

A partial order of sub-circuits arises by inclusion; minimal "atoms" might correspond to irreducible logic rules or morphological transformations.

7.48 Possibly Thousands or Millions of Atoms

Large models likely have a vast combinatorial array of sub-circuits.

7.49 Engineering Heuristics

- Start with broad tasks, identify large sub-circuits.
- Drill down on specialized tasks, isolating smaller sub-circuits.
- Build a lattice structure of bridging parameters.

7.50 Potential Gains in Robustness

A network with many well-defined small sub-circuits may degrade gracefully under random ablations.

7.51 Hypothesis—Why Overfitting Is Limited in Larger Models

A tangle of cohomological constraints among sub-circuits acts as a hidden regularization, explaining why giant models do not always overfit.

7.52 Unresolved Complexity—Dynamic Sub-circuit Evolution

Sub-circuits may merge or refine across training stages, so a single set might not remain stable throughout.

7.53 Infinity-Categorical Generalization

One could treat parameter updates as morphisms in an ∞-category, capturing higher homotopies (very speculative).

7.54 Slogan—Large NNs as Emergent Derived Categories

In derived algebraic geometry, we have derived categories of cochain complexes. Possibly large NNs form "algorithmic derived categories."

7.55 Potential Collaboration with Algebraic Geometry Community

Mathematicians could formalize these heuristics, bridging homological algebra and deep learning in a rigorous subfield.

7.56 Pragmatic Takeaway—High Risk, High Reward

Even a partial success in systematically leveraging exact sub-circuits could boost interpretability, robustness, and compression.

7.57 Proposed Next Steps

1. Implement sub-circuit mapping in a medium-scale model (e.g. 1B parameters).
2. Check stability under continued training.
3. Attempt partial "exactness-preserving distillation."

7.58 Potential Obstacles

- Modern LLMs are extremely large.
- Many arguments rely on linear approximations to parameter perturbations.
- Real chain complexes might be too messy in practice.

7.59 Nonetheless, Theoretical Beauty

Neural networks plus sheaf cohomology echo how advanced geometry found applications in theoretical physics.

7.60 Encouragement for Deeper Inquiry

Pushing these ideas might reorder common AI practices under a more rigorous framework.

7.61 Final Word on Long Exact Sequences

They track "what is lost" or "what is gained" at each representational layer, akin to features vanishing/appearing during training.

7.62 Final Word on Local-to-Global Spectral Sequences

They illustrate how partial coverage of the data domain can unify into a globally consistent learned model.

7.63 Aspiration

We aim to build "cohomological transformers" that incorporate these constraints systematically, offering interpretability, multi-task stability, and efficient distillation.

7.64 A Plea for Mathematical Rigor

Bridging continuous parameter realms, modern hardware, and approximate computations is nontrivial but worth exploring.

7.65 Conclusion

Despite the speculative nature, exploring sheaf cohomology, exact sequences, and homological invariants in neural networks might yield a powerful unifying framework for interpretability, training stability, and compression strategies.