AutoCircuit: Automated Discovery of Interpretable Reasoning Patterns in Large Language Models

Summary

This project aims to systematically discover interpretable reasoning circuits in large language models, by data mining attribution graphs from Neuronpedia's circuit tracer which is based on Anthropic's circuit tracing publication (https://transformer-circuits.pub/2025/attribution-graphs/methods.html). While the transformer circuits work demonstrates how to generate attribution graphs for individual prompts, manually analyzing thousands of graphs to identify common computational patterns is impractical.

Our approach will use LLM agents to automatically collect, process, and analyze attribution graphs across diverse prompt categories (factual recall, arithmetic, linguistic reasoning, etc.). The system will identify recurring subgraph patterns that represent stable computational circuits—reusable reasoning pathways that models consistently employ across similar tasks.

Key components include: (1) automated graph collection via Neuronpedia's API across systematically varied prompts, (2) graph simplification algorithms to extract core computational structures while filtering noise, (3) pattern recognition to identify circuit motifs that appear across multiple contexts, and (4) validation through targeted interventions on discovered circuits. The output will be a curated library of interpretable reasoning circuits with evidence for their causal role in model behavior, advancing our understanding of how LLMs actually think and enabling more targeted model analysis and alignment research.

Extended project description

Theory of Change

Automated circuit discovery could significantly contribute to reducing AGI risks by democratizing mechanistic interpretability and enabling real-time safety monitoring. Currently, understanding transformer internals requires extensive manual analysis, limiting interpretability research to small teams of specialists. By automating feature annotation, circuit hypothesis generation, and validation processes, automated circuit discovery would enable rapid identification of dangerous capabilities before they cause harm. Automated systems could continuously monitor deployed models for emerging deceptive behaviors, escape-seeking patterns, or capability jumps that might indicate misalignment. Ultimately, the goal in this field is to scale interpretability research from analyzing individual circuits to mapping entire LLM model cognitive architectures, enabling proactive safety measures rather than reactive responses. An issue of concern with agents would be bias due to the model training that is driving the agent. Our approach would be to manually confirm the circuit selections by agents, using a subset of the analyzed circuits, selected based on a metric such as the graph structure (centrality, distance between and number of pruned nodes by the agent etc). Furthermore, automated circuit discovery could accelerate AI alignment research by providing systematic understanding of how models represent goals, values, and decision-making processes, enabling targeted interventions to ensure beneficial outcomes.

Key Assumptions

This theory of change assumes that AGI systems will continue using transformer-like architectures where mechanistic interpretability remains feasible, rather than shifting to completely opaque paradigms. It presumes that dangerous AI behaviors correspond to identifiable computational circuits that can be detected through automated analysis before causing irreversible harm. The approach requires that human society maintains sufficient coordination to implement interpretability-based safety measures, including regulatory frameworks that mandate circuit analysis for high-stakes AI deployments. Critical assumptions include that automated interpretability tools will be adopted by AI developers rather than being relegated to academic research, as currently the pace of circuit discovery and safety methodologies AI alignment, is than AI capability advancement. The theory also assumes that interpretability insights will translate into effective safety measures, rather than merely providing post-hoc explanations of already-occurred model unsafe or malicious behaviors. Success for large-scale deployment of this approach, depends on sufficient computational resources being available for real-time circuit analysis of increasingly large models.

Project Plan

Backup Plans

Primary Risk: Automated circuit discovery systems might generate numerous false positive circuit discoveries, overwhelming researchers with incorrect interpretations. The backup plan involves developing a range of validation methods that require multiple independent confirmation signals before accepting a circuit hypotheses, and implementing human-in-the-loop verification for safety-critical discoveries.
Technical Failure: If automated feature annotation proves insufficiently reliable, the project would pivot to semi-automated approaches that use AI systems to propose interpretations while requiring human validation. This maintains the research benefits while leading to accurate circuit predictions.
Scalability Issues: Should the approach fail to scale to larger models due to computational constraints, the backup involves developing targeted analysis methods that focus on safety-relevant circuit categories rather than comprehensive model analysis.

Project Scope

Phase 1: Automated Circuit Discovery and Feature Annotation

In this research project, we will implement automated feature interpretation by leveraging the cross-layer transcoder methodology and attribution graph construction algorithms from the circuit discovery framework published by Anthropic in 2025 (https://transformer-circuits.pub/2025/attribution-graphs/methods.html). We will employ language models to analyze activation patterns where features fire strongly, generating semantic interpretations that we validate through feature patching interventions. Our technical implementation will leverage the multiplicative steering capabilities demonstrated in the intervention demos and supported by Neuronpedia's model steering API functionality (https://www.neuronpedia.org/api-doc#tag/steering) we can systematically modify features through various intervention strategies including setting features to zero, amplifying their activations, or applying multiplicative scaling. These steering capabilities enable us to test causal hypotheses about feature function by observing how modifications propagate through the computational graph to affect downstream activations and final model outputs. Through Neuronpedia's graph visualization platform, we will also validate feature interpretations by demonstrating that interventions on semantically labeled features producing predictable and interpretable changes in model behavior, such as language

switching when modifying language-specific features or topic changes when steering content-related circuits. We will incorporate the display utilities for token predictions to visualize intervention effects, and build comprehensive databases of validated feature interpretations with confidence scoring based on intervention consistency and downstream effect measurements.

Phase 2: Systematic Circuit Validation and Exploration

We will systematically mine attribution graphs by analyzing the indirect influence computations. This will enable us to identify multi-step causal chains where features in early layers affect downstream computations through intermediate feature activations, revealing hierarchical circuit structures that implement complex behaviors. Our graph mining methodology will be automated for the most part using an LLM (Claude Sonnet), that can analyze the adjacency matrix patterns and propose hypotheses about which feature combinations form coherent computational circuits. The LLM will also interpret activation co-occurrence patterns and suggest semantic groupings based on the direct effect measurements between nodes. Our automated approach will focus on detecting feature clusters that consistently co-activate across related prompts, using the direct effect measurements encoded in the attribution graph's adjacency matrix. This approach enables us to quantify the strength of feature-to-feature interactions and identify computational modules that work together to implement specific functions. Through this analysis, we will remove the redundant nodes which do not add explanatory value to understanding the model's output generation process. We will utilize advanced exploration techniques with the generation comparison utilities to test circuit modifications across extended sequences, ensuring that our discovered computational patterns generalize beyond single-token predictions. The LLM driving our circuit path analysis will receive continuous feedback through graph completeness and replacement scoring metrics, in addition to manually confirming the circuit selections by agents for graphs that are selected by the metrics. This will allow it to iteratively refine its circuit hypotheses and focus on the most explanatorily powerful computational pathways, while using these quantitative measures to guide its exploration and pruning decisions. We will leverage Neuronpedia's API endpoints for graph storage and visualization, enabling us to programmatically upload our generated and modified pruned graphs for interactive exploration and collaborative annotation through the platform's web interface.

Phase 3: Cross-Model Pattern Analysis and Deployment

Using circuit-tracer's ReplacementModel framework, we can load different model architectures with their corresponding transcoder configurations and generate attribution graphs for identical prompts, then systematically compare the resulting adjacency matrices to identify structurally similar computational pathways. Through Neuronpedia's graph storage and visualization capabilities, we can upload these cross-model attribution graphs and leverage the platform's annotation (http://neuronpedia.org/gemma-2-2b/graph) tools to manually validate that circuits with similar graph structures actually implement the same semantic functions and respond similarly to prompts. Furthermore, we can test this using the interactive steering interface to test whether interventions on corresponding features produce equivalent behavioral changes across different models. Our system will incorporate the graph completeness scoring and indirect influence matrix analysis to develop comparison metrics that account for architectural differences while identifying universal computational patterns. The real-time circuit monitoring for our deployment framework will be implemented as an extension to the existing circuit-tracer functionality, building upon the attribution computation pipeline to continuously analyze feature activation patterns and computational pathway changes in deployed models. We will extend the current batch processing and graph generation capabilities to support streaming analysis of model behavior, implementing automated alerts that trigger when significant deviations from baseline circuit patterns are detected, indicating potential emergence of dangerous capabilities relevant to AI safety.

Included in Scope

- Automated annotation of model features using attribution graph analysis
- Systematic circuit discovery and hypothesis generation methodologies
- Validation frameworks for testing LLM model computation hypothesis through mechanistic interventions
- Cross-model comparison techniques for identifying universal safety-relevant patterns
- Integration with existing interpretability infrastructure including Neuronpedia and circuit-tracer frameworks

Excluded from Scope

- Development of new transcoder training methodologies or fundamental interpretability techniques
- Creation of novel model architectures designed for interpretability
- Regulatory policy development or implementation of industry safety standards
- Analysis of non-transformer architectures or fundamentally different AI paradigms

Most Ambitious Version

A comprehensive automated interpretability platform that can continuously monitor deployed AI systems for emerging dangerous capabilities, automatically identify and validate safety-relevant circuits in real-time, and provide actionable interventions to prevent harmful behaviors before they manifest. This would include automated generation of safety benchmarks, real-time circuit analysis during model training, and integration with AI development pipelines to enable interpretability-guided model design. While this might be too ambitious for the time limits of this project, our open source code can be used as a basis for others to build it out to its full potential.

Least Ambitious Version

A suite of semi-automated tools that accelerate existing manual interpretability research by providing AI-assisted feature annotation and circuit hypothesis generation. This minimal version would primarily serve as a research accelerator for interpretability specialists, reducing the time required for manual circuit analysis while maintaining human oversight for all critical safety determinations. The tools would integrate with the existing framework of Neuronpedia, where at minimum a collection of circuits found through this project will be published.

Output

All circuits discovered in the proposed project will be published on Neuronpedia, and all code developed will be placed on Github with open source license. We will also do an arxiv paper which will also be submitted prior to a conference (ex. NeuroIPS 2026).

Risks and downsides

No risks other than mis-interpreting circuits, but the safeguards are built in the research methodology as described in Phase 1-3 as presented in the earlier section of the document.

Team

Team size

3-5 people total, flexible to work on EST or CET time zone depending on the majority of the group, The lead and people who join the project are expected to spend a minimum of 10 hours per week on this project during its official duration.

Project Lead

Konstantinos Krampis
https://kkrampis.github.io/blog/curriculum-vitae/index.html

Skill requirements

Experience coding with Python,understanding APIs and graph data structures, ideally having run TransformerLens (https://transformerlensorg.github.io/TransformerLens/) or ARENA AI safety workshop materials which are available online (https://arena-chapter1-transformer-interp.streamlit.app/).

Knowing clearly the Transformer LLM architecture, having read (and clearly understood) the Antropic papers (https://transformer-circuits.pub/), Neel Nanda's excellent materials would also get you quickly up to speed https://www.neelnanda.io/mechanistic-interpretability/quickstart-old.