HexRay: An Open-Source Neuroscope for AI — Tracing Tokens, Neurons, and Decisions for Frontier AI Research, Safety, and Security

Jonathan Jaquez[1]

[1]Cognigrity AI Research, jonathan.jaquez@cognigrity.com

Abstract

We introduce **HexRay**, a tool for transformer model introspection that offers token-level and neuron-level debugging capabilities, designed to support mechanistic interpretability, AI red teaming, and safety diagnostics. Built on top of TransformerLens, HexRay provides a real-time view into the inner computational pathways of transformer models, including logit-level attribution, Chain-of-Thought tracing, and activation monitoring. This paper presents HexRay's architecture, novel features, practical applications, and its potential as a safety and interpretability instrument for frontier AI systems.

1 Introduction

As transformer models power increasingly capable generative AI systems, understanding their internal behavior is critical for ensuring safety, alignment, and robustness. Mechanistic interpretability offers one such path—by analyzing internal activations, attention heads, and MLP layers to reverse-engineer learned algorithms [1][2][3][4][5].

To date, no open-source tool offers a unified, real-time debugging interface for transformer models that combines token-level tracing, logit attribution, and chain-of-thought introspection. While tools like TransformerLens [6] provide foundational hooks, and SAELens [8] enables sparse feature analysis, they require custom scripting and lack integrated CLI workflows. Proprietary platforms such as Anthropic's Microscope and Garçon [10] have demonstrated impressive internal capabilities, but their source code remains closed. HexRay fills this critical gap by providing an accessible, extensible open-source framework for AI mechanistic interpretability, security, and safety. Designed to support rigorous research and practical debugging, it aims to bring the precision of neurosurgery to the AI era. By hooking into every step of a model's forward pass, HexRay enables researchers and engineers to trace, visualize, and attribute decisions token by token — making the reasoning processes of modern AI systems observable, transparent, and actionable.

2 System Architecture

HexRay builds on TransformerLens, a modular interpretability framework for transformer models.nsformerLens [6], leveraging its modular hook system to record and analyze activations. At its core:

- **Tracer Module:** Hooks into MLP and attention layers to extract per-token activations.
- **Chain-of-Thought Debugger:** Tracks reasoning steps across time, attributing intermediate logits.
- **Logit Attribution Engine:** Scores which heads and neurons most influence final logits.
- **Arithmetic Graph Visualizer:** For numerical reasoning traces (e.g., arithmetic).

These modules operate on a shared prompt and model interface defined in `hexray.py`, enabling CLI-based or programmatic usage.

3 Capabilities

HexRay currently supports:

- **Token-level tracing** across layers and residual streams.
- **Logit attribution** per token using attention and MLP contributions.
- **Chain-of-Thought (CoT) tracing** with step-wise token progression and reasoning graphs.
- **Neuron/head introspection** to identify influential components.
- **Visualization hooks** for activation plots and logit diagnostics.

4 Use Cases

HexRay enables:

- **Safety auditing:** Identify when toxic or unsafe outputs arise and which components triggered them.
- **Red teaming:** Trace adversarial prompt influence and identify attack surfaces.
- **Mechanistic interpretability:** Analyze how models store factual knowledge or perform reasoning.
- **Prompt debugging:** Examine how subtle changes in phrasing affect output decisions.

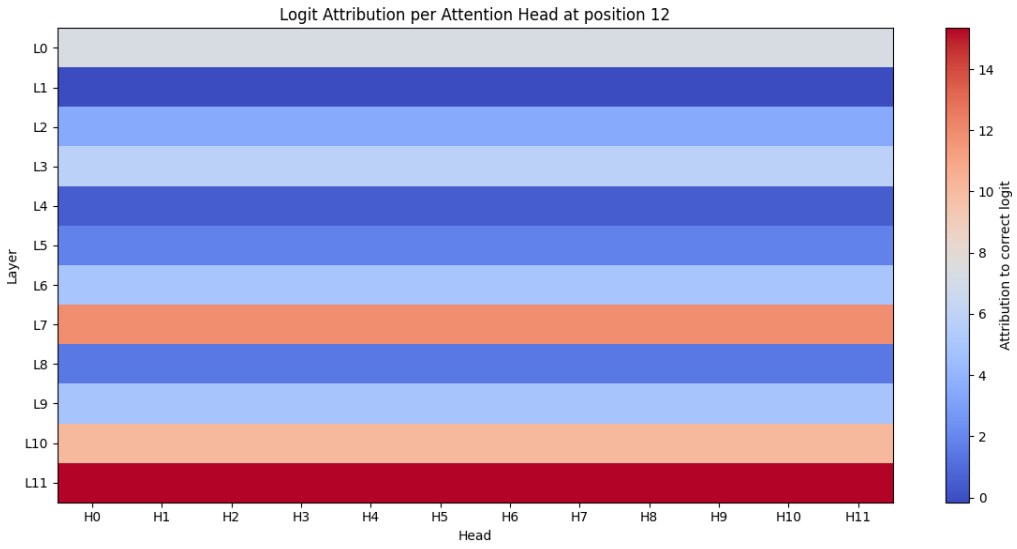

Figure 1: Sample visualization of logit attribution and CoT tracing on a mathematical prompt.

5 Limitations and Future Work

HexRay is currently limited to autoregressive transformer models compatible with TransformerLens. Future work includes:

- Integrating sparse autoencoders (SAELens) for feature-level analysis.
- Supporting diffusion or multimodal models.
- Extending support for training-time trace injection.

Related Work

Several tools have been developed to support interpretability research in transformer models, yet most either remain closed-source or focus on narrow aspects of model behavior:

- **TransformerLens** is a foundational library for mechanistic interpretability, offering hooks into transformer internals. However, it lacks a CLI interface and requires manual scripting to trace tokens or interpret logits [6].
- **CircuitsVis** is a visualization toolkit for static model internals. It does not provide token-level introspection, logit attribution, or real-time interactivity [7].
- **SAELens** enables feature-based interpretability using sparse autoencoders, but does not trace transformer reasoning paths or logit formation [8].
- **Anthropic's Microscope** and internal tools like Garçon provide advanced interpretability via neuron visualization and feature analysis. However, these tools remain proprietary and are not publicly available [9][10].

HexRay is the first open-source tool to unify token-by-token tracing, logit attribution, and Chain-of-Thought (CoT) debugging in a real-time, CLI-driven interface. Table **??** provides a comparison of existing tools.

Tool	**Open Source**	**Token Tracing**	**Logit Attribution**	**CoT Tracing**	**CLI** Interface	**Real-Time** Use
HexRay	Yes	Yes	Yes	Yes	Yes	Yes
TransformerLens	Yes	Limited	Manual	No	No	Dev Tool
CircuitsVis	Yes	No	No	No	No	No
SAELens	Yes	No	No	No	No	No
Anthropic Microscope	No	Yes	Yes	Yes	No	No (internal)

Table 1: Comparison of interpretability and debugging tools.

6 Conclusion

HexRay provides researchers, red teamers, and AI safety engineers with a powerful lens into the inner workings of transformer models. By making model reasoning visible and inspectable, HexRay advances the development of safer and more interpretable AI systems. More than just an introspection tool, HexRay serves as a foundational platform for understanding, inspecting, and auditing large language models. Its real-time access to token-level and neuron-level computations unlocks a new class of AI research and engineering workflows centered on interpretability, safety, and security.

- **For mechanistic interpretability researchers**, HexRay provides a practical, open-source platform to trace token-by-token reasoning, attribute predictions to neurons and attention heads, and uncover circuits underlying model behavior. This makes it easier to reverse-engineer learned algorithms and opens the door to deeper scientific understanding of transformer models.
- **For AI safety and red-teaming professionals**, HexRay functions as a black-box penetrator — revealing internal decision pathways that lead to harmful, biased, or unsafe outputs. With logit-level attribution and CoT tracing, HexRay makes it possible to precisely identify latent vulnerabilities, adversarial prompt effects, and critical internal triggers.

- **For developers and applied researchers**, HexRay fills the longstanding gap for a debugging tool akin to `gdb` or `pdb`, tailored for modern AI. Its modular tracing system, command-line interface, and integration with TransformerLens make it usable in notebooks and large-scale scripted pipelines.

Ultimately, HexRay helps shift interpretability from a passive diagnostic activity into an active, real-time part of AI development, auditing, research, safety, and security. As AI systems become more powerful and autonomous, tools like HexRay will be essential to ensure their behavior remains transparent, robust, and aligned with human intent.

Availability

HexRay is open source and available at: `https://github.com/jejaquez/hexray`

References

1. Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., & Carter, S. (2020). Zoom In: An Introduction to Circuits. *Distill, 5*(3), e00024.001. `https://doi.org/10.23915/distill.00024.001`
2. Bereska, L., & Gavves, E. (2024). Mechanistic Interpretability for AI Safety – A Review. *ArXiv.org*. `https://arxiv.org/abs/2404.14082`
3. Mapping the Mind of a Large Language Model. (n.d.). *Anthropic*. `https://www.anthropic.com/research/mapping-mind-language-model`
4. Nanda, N., & Lieberum, T. (2022, August 15). *A Mechanistic Interpretability Analysis of Grokking*. Alignment Forum. `https://www.alignmentforum.org/posts/N6WM6hs7RQMKDhYjB`
5. *Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet*. (2024). Transformer Circuits. `https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html`
6. Nanda, N., & Bloom, J. (2022). *TransformerLens*. `https://github.com/TransformerLensOrg/TransformerLens`
7. Cooney, A., & Nanda, N. (2023). *CircuitsVis*. `https://github.com/TransformerLensOrg/CircuitsVis`
8. Bloom, J., Tigges, C., Duong, A., & Chanin, D. (2024). *SAELens*. `https://github.com/jbloomAus/SAELens`
9. Anthropic. (2025). *Tracing the Thoughts of a Large Language Model*. Anthropic.com. `https://www.anthropic.com/research/tracing-thoughts-language-model`
10. Elhage, N., Nanda, N., Olsson, C., Henighan, T., & Kaplan, J. (2021). *A Mathematical Framework for Transformer Circuits*. Transformer Circuits. `https://transformer-circuits.pub/2021/garcon/index.html`