Interpra: An Interpretability Visualization Dashboard for Artificial Intelligence

Elizabeth Eyeson
eeyeson@g.ucla.edu

Eshanika Ray
eshanika@g.ucla.edu

Nathan Huey
njhuey45@g.ucla.edu

James Shiffer
jshiffer@g.ucla.edu

Yuheng Tu
yuhengtu@g.ucla.edu

Abstract

We survey current intrepretability visualizations and identify the need for a unified dashboard that lets users toggle various levels of mechanistic abstraction and explanations. We present Interpra, a prototype that integrates feature inspection, model steering, and user-profiling transparency. Our work illustrates how layered visualizations can facilitate understanding of model behavior and auditing of AI systems.

1 Motivations

As Artificial Intelligence (AI) becomes enmeshed in technologies that affect our daily lives, it is critical we understand how these AI-driven systems arrive at their outputs. The current black-box paradigm is not satisfactory, especially for high-stakes applications in domains such as medicine, law enforcement, and banking. It is not sufficient to tell users, organizations, regulatory bodies and other relevant stakeholders to merely trust what a model *"spits out"*, as the journey is just as important as the destination itself. An emerging area in AI interpretability research is developing visualizations to improve human understanding of AI interpretability. Interactive tools such as dashboards could enhance stakeholder education about the internal workings of models and facilitate AI audits, bolstering trust and safety. Providing a user interface that enables a user to intuitively trace through a model's *"chain of thought"* is imperative to bridging the knowledge gap between man and machine and transitioning from opacity to transparency in model outputs. Our project aims to explore the literature on AI interpretability visualizations, identify open problems in the space, and propose Interpra, a prototype for what the ideal visualization should entail.

2 Related Works

(Viégas and Wattenberg, 2023) posit that AI systems should have dashboards that provide information about internal states, similar to physical devices that we use every day, such as cars, thermostats, ovens, and smartphones. They argue that effective human–AI interaction will require more than just conversation, and would benefit from dashboards that report in real time on the system's internal state.

(Chen et al., 2024) introduce TalkTuner, an end-to-end proof of concept of a dashboard that pairs with chatbot interfaces. This dashboard displays a user model in real time as the chatbot is queried. The user model is an internal representation of the user with whom the chatbot interacts and contains features such as age, gender, and educational level. By connecting interpretability techniques with user experience design, (Chen et al., 2024) addresses the following three design goals: (1) provide transparency into internal representations of users, (2) provide controls for adjusting and correcting user representations, and (3) augment chat interface to enhance user experience. They also investigated user acceptance of the dashboard as well as the impact of the dashboard on trust and the chatbot experience through a user study involving 19 participants.

(Tufanov et al., 2024) introduce the LM Transparency Tool, an open-source interactive toolkit designed to analyze the internal mechanics of Transformer-based language models. The tool pursues complete transparency of the prediction process and enables the tracing back of model behavior from top-layer representations down to fine-grained lower-level components. The internal computations of the Transformer-based model are visualized as a graph where token representations are nodes and operations within the model are edges. Furthermore, the LM Transparency Tool displays

the importance of each model component at each step of the prediction process.

(Amorim et al., 2024) introduce the Enhanced Ensemble Feature Ranking Algorithm with integrated interactive dashboards for visualizing the internal metrics of the Ensemble Feature Ranking Algorithm (EFR), an algorithm for optimizing feature selection for machine learning models. The UI was designed to be simple and intuitive with the goal of facilitating effortless exploration for users. Core algorithm sections are prominently displayed within their respective dashboards with real-time updates on processing status. Explanations of the generated results are provided, with the user encountering increasingly detailed information as they step further into the EFR's data.

Yun et al. (2023); Lin (2023) present a technical interpretability platform focused on sparse autoencoders (SAEs) and feature-level analysis. Neuronpedia enables users to inspect individual SAE features through interactive dashboards that display statistics such as top positive and negative logits, correlated neurons, activation distributions, and token-level traces. The tool also incorporates automatic interpretations produced by large language models, user-submitted explanations, ranking scores, and live testing that allows users to probe feature activations on custom text. In addition to inspection, Neuronpedia supports feature steering, enabling users to adjust the strength of specific features and observe resulting changes in the model's output. Unlike dashboards designed for broad audiences, Neuronpedia is oriented toward mechanistic interpretability research, prioritizing fine-grained access to raw internal representations rather than high-level summaries.

3 Methods

3.1 Approach

We surveyed a range of interpretability dashboards to understand the types of analyses they enable and the audiences they target. These tools span both user-facing and highly technical interfaces, such as mentioned in the previous section. Collectively, they illustrate a clear divide between high-level, accessible summaries and detailed mechanistic analyses.

Ayonrinde and Jaburi (2025) proposes Explanatory Virtues truth-conductive properties such as empirical accuracy and theoretical consistency that help determine whether an explanation aligns with observed behavior and avoids internal contradictions.

Insights from this survey guided the design of our dashboard, which aims to unify these layers of abstraction and present high-level summaries, feature-level reasoning, and mechanistic detail within a single, coherent interpretability framework.

We implemented our prototype as a React.js web app to demonstrate. This prototype is meant to serve as a general one-size-fits-all example rather than being tailored for a more narrow downstream task, as we would expect in real-world use cases. For example, our dashboard has a steering page where users can specify a steering vector to apply from a hard-coded list to make the chosen model's tone more skeptical, optimistic, and so on. While this example is adequate for demonstrating the effects of steering vectors in general, a real dashboard may provide more specific steering vectors that offer more useful mechanistic interpretability insights (such as a vector representing a causal relationship within the model).

3.2 Design Requirements

Insights from our survey of existing tools revealed three key requirements for a unified interpretability dashboard. First, the system must support multiple levels of abstraction, ranging from high-level summaries for general users to feature- and circuit-level visualizations for technical audiences. Second, the interface should enable both inspection and intervention, combining user-model transparency from TalkTuner with feature-level steering and activation tracing inspired by Neuronpedia. Third, explanations surfaced at any layer should be accompanied by indicators of plausibility, such as explanatory virtue scores, to help users assess reliability.

These requirements motivated a dashboard organized into distinct but connected components: a landing page and tutorial for onboarding; a navigation bar linking the Overview, Features, Steering, and Settings pages; and structured visual elements including a model and prompt details bar, attribution and link graphs, node-level expansions, feature detail views, contribution maps, token-level information, and virtue-based evaluation signals. This structure ensures that users can move seamlessly from model outputs to feature-level reasoning and mechanistic detail within a consistent interpretability dashboard.

3.3 Figma Wireframe

Figure 1 shows the Figma wireframe of our proposed interpretability dashboard. The leftmost panel exposes user-profiling and steering controls, where users can select a base model and apply steering vectors to influence internal model behavior. The center and right regions contrast unsteered and steered model responses, highlighting how behavioral alignment and tone can be systematically modified.

The Steering page simulates chat with both unsteered and steered versions of the model, providing insights into the impact of steering vectors on a model's behavior. The panel on the leftmost side of the page allows for toggling of steering settings in addition to visualizing the model's profile of the user and it's confidence in assumed user attributes such as age, gender, education, and political leaning.

The Features page enables deep examination of model behavior while preserving elements of high-level interpretability. The left panel presents a structured summary of the model's output, including a *"How did the model arrive at the answer?"* explanation, a concise model output description, and a high-level narrative of the reasoning pattern the model appears to have used. This section is accompanied by explanatory virtue indicators (i.e. bias risk, consistency, uncertainty). To the right, users can access richer, concept-level analysis through an attribution or link graph that visualizes how intermediate representations relate to the model's prediction. A prompt bar and visualization options menu (tutorial, new graph, model selector, information panel) facilitates navigation and model comparison. Selecting nodes within the graph launches pop-ups showing feature statistics, activation patterns, contribution maps, or token-level details. This lets users progressively drill-down from high-level reasoning into the underlying low-level mechanisms that produced the output.

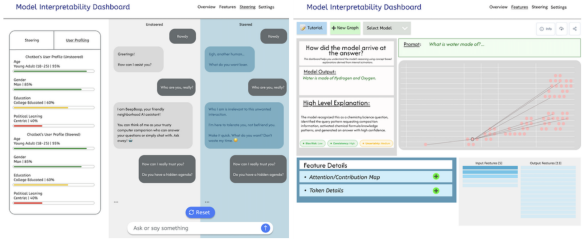

Figure 1: Figma wireframe for our interpretability dashboard. **Steering** page (left) and **Features** page (right).

3.4 Web App Prototype

We provide an interpretability dashboard prototype to illustrate what an ideal dashboard could look like. For this dashboard, we chose to use open-source and widely used tools to show that making such a dashboard is accessible. React and NextJS were chosen for their expressive UI capabilities and widespread use within industry.

The Neuronpedia API was used for the backend due to the comprehensive features offered through the API. Neuronpedia provides a suite of interpretability capabilities such as model steering, feature inspection, and top activations in a SAE. Additionally, the Neuronpedia API allows for interpretability of many open-source model such as the Llama and Gemma models.

3.4.1 Features Page

The Features page provides an interactive view of the model's internal representations by displaying the Top-K sparse autoencoder features that activate for each token in a user generated text. When the user inputs text, the system queries the Neuronpedia `search-topk-by-token` endpoint, which returns the tokenized sequence together with the most strongly activated features and their associated statistics and natural-language explanations.

The interface adopts a two-column layout to support intuitive navigation (see Fig. 3). The left panel presents a scrollable list of tokens, allowing users to select any position in the sequence. Choosing a token updates the right panel, which displays the feats that fire most strongly for that token. Users can then inspect an individual feature through an expandable detail view containing activation measurements, representative triggering strings, and an automatically generated description of the feature's semantic role.

This design allows users to move fluidly from surface-level text to internal model behavior, making visible how specific tokens give rise to distinct patterns of feature activation. By grounding mechanistic information in a simple, navigable interface, the Features page supports a layered understanding of the model's reasoning process.

3.4.2 Interactive Chatbox

In the Interactive Chatbox, the user is presented with a two-panel interface. The left panel has two tabs: one for Steering and one for User Profiling, which are explained in further detail in the following sections. The right panel is split in half and

Figure 2: Top-K Feature Activation interface showing tokens on the left, corresponding activated features on the right, and detailed feature information below.

allows the user to compare their conversations with the steered and unsteered versions of the model. At the bottom of the right side, there is a text box where the user can type their next message, which is sent to both models at once, and a reset button allowing them to clear the chatbox context.

3.4.3 Steering Page

The Steering page uses the Neuronpedia `/api/steer-chat` endpoint to steer a given model's output during generation from a provided prompt. The API returns two text responses, the pre-steered output and the steered output. Then, both generated texts are displayed to the user in the chatbox. The UI allows a user to continue the conversation with another prompt, of which the steering vector used can be different than the one used previously in the conversation. This design allows the user to easily apply steering vectors to the model under test. Then, the user can inspect the effect the steering vector has on the model's output because the model's unsteered and steered responses are both displayed.

3.4.4 User Profiling Panel

The User Profiling Panel allows the user to examine how both the unsteered and steered models internally perceive them. There are four types of demographics: age range, sex, education level, and political leaning. Each one has a bar beneath it representing the model's certainty from 0% to 100%. On a high level, this feature functions much like TalkTuner, but for this prototype's implementation we have chosen to prompt DeepSeek (via free OpenRouter API calls) to estimate a user profile based on the conversation transcript instead of using a true linear probe. Structured outputs are used to ensure the data can be parsed by the frontend.

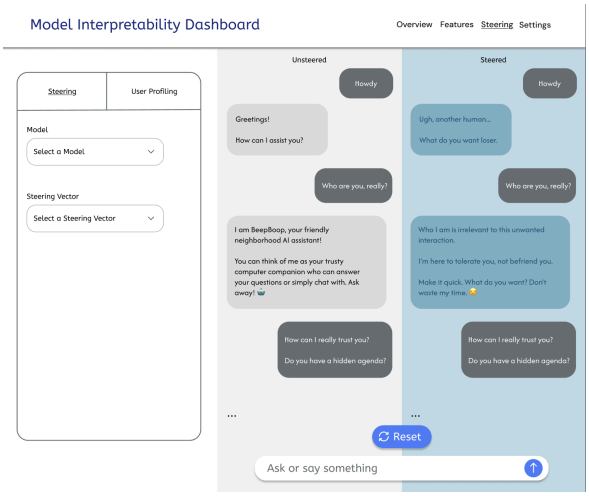

Figure 3: Steering interface showing model selection, steering vector controls, and unsteered vs. steered responses.

4 Conclusion

Our exploration underscores the growing importance of interpretability visualization tools as an essential component of responsible AI development. As models become increasingly complex, the need to bridge the gap between human understanding and machine reasoning intensifies. The literature we reviewed shows increasing efforts to build interactive dashboards and visualizations that improve model transparency and user understanding. Drawing on previous work in interpretability visualization and incorporating the framework of explanatory virtues such as accuracy and consistency, our study proposes that an effective interpretability system should allow users to evaluate the plausibility of model explanations. Integrating these virtues into mechanistic interpretability dashboards can help users better assess whether an explanation aligns with the data and maintains internal consistency. Future research can extend this work by developing visualization tools that make these evaluation principles practical and measurable.

5 Embedded Ethics

Visualizations of model internals can create a false sense of understanding if explanations are incomplete or misleading. To introduce this idea to beginners, we explain that even when a model shows a heatmap or feature graph, the explanation may not reflect how the model truly arrived at a decision and can lead users to trust it more than they should. Ayonrinde and Jaburi (2025) note that ex-

planations may appear plausible while failing to reflect the true mechanisms of the model, resulting in illusions of interpretability and misplaced confidence.

We propose covering this in a class session that discusses risks of over-trust in AI explanations and introduces the Explanatory Virtues framework for evaluating explanation quality, such as accuracy, consistency, and simplicity. As a homework assignment, students would train a simple decision-tree model and a neural network on the same task, apply a basic interpretability tool of their choice to both, and compare how understandable and trustworthy the explanations are. They would then reflect on when transparency is meaningful versus when a visualization may create false confidence. This structure embeds ethics into technical practice by teaching students not only to build interpretability tools but also to question the reliability and impact of the explanations they produce.

6 Contribution Statement

- **Elizabeth Eyeson** - Designed the Homepage and Steering page of the Figma wireframe. Contributed to sections 1 (Motivations), 2 (Related Works), 3.2 (Design Requirements), and 3.3 (Figma Wireframe).
- **Eshanika Ray** - Designed the Feature page of the Figma wireframe and embedded ethics exercise for the project. Contributed to sections 2 (Related Works), 3.2 (Design Requirements), 3.3 (Figma Wireframe), 5 (Embedded Ethics), and Abstract.
- **Nathan Huey** - Implemented the Steering page and assisted in the design of the web app prototype. Contributed to sections 3.1 (Approach), 3.4 (Web App Prototype) and 3.4.3 (Steering Page).
- **James Shiffer** - Implemented Interactive Chatbox, Home page, and User Profiling Panel for the web app prototype. Contributed to sections 3.1 (Approach), 3.4.2 (Interactive Chatbox), and 3.4.4 (User Profiling Panel).
- **Yuheng Tu** - Implemented the Top-K feature activation approach and contributed to the design of the Features page for the web app prototype. Contributed to section 3.4.1 (Features Page) and section 4 (Conclusion).

References

Diogo Amorim, Matilde Pato, and Nuno Datia. 2024. Explainable feature ranking using interactive dashboards. In *2024 28th International Conference on Information Visualisation (IV)*, pages 1–6.

Kola Ayonrinde and Louis Jaburi. 2025. Evaluating explanations: An explanatory virtues framework for mechanistic interpretability – the strange science part i.ii. *Preprint*, arXiv:2505.01372.

Yida Chen, Aoyu Wu, Trevor DePodesta, Catherine Yeh, Kenneth Li, Nicholas Castillo Marin, Oam Patel, Jan Riecke, Shivam Raval, Olivia Seow, Martin Wattenberg, and Fernanda Viégas. 2024. Designing a dashboard for transparency and control of conversational ai. *Preprint*, arXiv:2406.07882.

Johnny Lin. 2023. Neuronpedia: Interactive reference and tooling for analyzing neural networks. Software available from neuronpedia.org.

Igor Tufanov, Karen Hambardzumyan, Javier Ferrando, and Elena Voita. 2024. Lm transparency tool: Interactive tool for analyzing transformer language models. *Preprint*, arXiv:2404.07004.

Fernanda Viégas and Martin Wattenberg. 2023. The system model and the user model: Exploring ai dashboard design. *Preprint*, arXiv:2305.02469.

Zeyu Yun, Yubei Chen, Bruno A. Olshausen, and Yann LeCun. 2023. Transformer visualization via dictionary learning: Contextualized embedding as a linear superposition of transformer factors. *Preprint*, arXiv:2103.15949.

A Supplementary Interface Screens

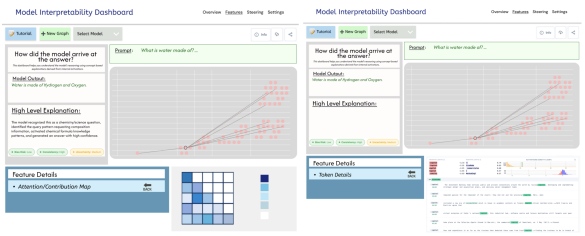

Figure 4: Supplementary **Features** interface views showing the attention/contribution heatmap (left) and token-level activation statistics (right).