Interpretabilidade em Grandes Modelos de Linguagem

1st Mariana Assis Ramos
Departamento de Sistemas de Informação (DCC)
Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte, Brasil
mariana.assis@dcc.ufmg.br

***Abstract*—A consolidação dos Grandes Modelos de Linguagem (LLMs), exemplificada pelo GPT-4 [1] e Gemini 3 [2], ampliou drasticamente a capacidade da IA, mas agravou proporcionalmente o problema da "caixa-preta". Embora a macroestrutura de modelos autoregressivos seja conhecida, os mecanismos algorítmicos internos permanecem opacos devido a fenômenos emergentes como a polissemia neural e a superposição [3]. A literatura atual fragmenta-se entre paradigmas de treinamento (fine-tuning vs. prompting) [4] ou limita-se à dicotomia clássica de acesso aos parâmetros (black-box vs. white-box) [5]. Tais abordagens mostram-se insuficientes para capturar a profundidade cognitiva e o propósito epistêmico das novas metodologias. Este trabalho sistematiza o estado da arte ao propor uma Taxonomia Multidimensional estruturada em dois eixos: (1) Nível de Abstração Cognitiva, distinguindo métodos Computacionais, Representacionais e Mecanicistas; e (2) Propósito Epistêmico, diferenciando abordagens Descritivas, Causais, Intervencionais e Verificáveis. Sob esta ótica, reavaliamos métodos como Sparse Autoencoders e Induction Heads, demonstrando a aplicabilidade da taxonomia através de um estudo de caso na plataforma Neuronpedia. Os resultados indicam uma transição da área para uma ciência intervencionista, complementar à engenharia de representações [6], focada não apenas em descrever, mas em controlar o comportamento do modelo.**

***Index Terms*—Grandes Modelos de Linguagem; Interpretabilidade Mecanicista; Taxonomia; Engenharia de Representação; Explainable AI.**

I. Introdução

A. Contexto e Motivação

Desde a publicação do artigo seminal *Attention Is All You Need* [7], que introduziu a arquitetura Transformer, o campo da inteligência artificial passou por uma transição paradigmática. A capacidade de treinar modelos de forma altamente paralela permitiu o surgimento dos Grandes Modelos de Linguagem (LLMs), que atualmente representam o estado da arte em raciocínio, geração de texto e multimodalidade. Esse avanço, consolidado pelo GPT-4 [1] e expandido por sistemas como o Gemini 3 [2], demonstra habilidades autônomas que vão da geração de código a análises complexas.

Apesar desses progressos, o aumento na capacidade dos modelos foi acompanhado por um crescimento proporcional em sua complexidade, intensificando o problema da "caixa-preta". Embora a macroestrutura dessas arquiteturas seja bem compreendida desde os primeiros modelos autoregressivos como o GPT-1 [8], os mecanismos algorítmicos que determinam a geração de uma resposta específica permanecem amplamente opacos. Esses modelos não são projetados manualmente; eles emergem do processo de treino. Se fossem construídos por engenheiros de forma explícita, interpretá-los seria trivial; porém, são sistemas adquiridos a partir de bilhões de exemplos, iterados trilhões de vezes, resultando em estruturas internas que nunca foram "desenhadas", apenas descobertas.

Essa opacidade agrava-se devido a fenômenos emergentes como a polissemia neural (*polysemanticity*), na qual a rede, ao precisar representar mais conceitos do que possui neurônios, recorre à superposição: múltiplas representações são comprimidas em um mesmo recurso computacional [3]. Isso impede explicações causais diretas e contribui para riscos reais, como alucinações, amplificação de vieses e dificuldades de auditoria.

Conforme argumentado por Lipton [9], não há uma definição consensual de interpretabilidade. Em vez disso, o termo abrange um conjunto de noções frequentemente complementares ou até conflitantes, unificadas pela necessidade de reduzir o descompasso entre o objetivo formal dos modelos (minimizar erro) e os objetivos humanos (segurança, ética, confiabilidade).

Nesse contexto, a interpretabilidade deixa de ser um luxo acadêmico e torna-se um requisito de segurança. Molnar [5], com base em Adadi e Berrada [10], sintetiza seu papel em três eixos fundamentais:

1) Depuração e diagnóstico;
2) Justificativa e confiabilidade;
3) Descoberta científica sobre o funcionamento interno.

B. Lacuna de Pesquisa

Embora o interesse em interpretabilidade tenha crescido exponencialmente, a área carece de estruturação conceitual quando aplicada a Grandes Modelos de Linguagem.

Trabalhos recentes, como Zhao et al. [4], organizam as técnicas considerando paradigmas de treinamento (*fine-tuned* vs. *prompting*). Essa classificação é útil sob a ótica do ciclo de vida do modelo, mas não captura adequadamente os diferentes propósitos epistemológicos e níveis de profundidade explicativa. Ressalta-se ainda que tal *survey* trata interpretabilidade e explicabilidade de forma intercambiável.

De forma semelhante, a distinção clássica de Molnar entre métodos *model-agnostic* e *model-specific* [5], aqui referidos

como *white-box* e *black-box*, já não é suficiente. Métodos *white-box* atuais podem ser:

- Descritivos: apenas visualizando pesos ou padrões; ou
- Mecanicistas: reconstruindo algoritmos internos.

Além disso, o *survey* mais completo sobre interpretabilidade interna [11] limita-se a DNNs tradicionais, não cobrindo as particularidades dos Transformers.

Surge, portanto, uma lacuna: faltam taxonomias que classifiquem as técnicas não apenas pelo tipo de acesso, mas pela natureza cognitiva da explicação (qual pergunta a técnica responde?) e pelo propósito científico (o que buscamos entender?). Em particular, é necessário diferenciar métodos que:

- Observam comportamentos externos;
- Extraem conceitos representacionais;
- Realizam engenharia reversa de circuitos;
- Intervêm ativamente no processo computacional.

C. Objetivo

O objetivo geral deste trabalho é realizar um *survey* sistemático sobre interpretabilidade em Grandes Modelos de Linguagem, propondo uma nova estrutura de classificação que organize o estado da arte de forma precisa e informativa.

A pergunta de pesquisa que orienta este trabalho é:

> *Como organizar as técnicas de interpretabilidade em uma estrutura unificada que capture tanto a profundidade cognitiva da explicação quanto o propósito epistêmico que ela busca atender?*

D. Contribuições

Para responder à pergunta de pesquisa, este trabalho oferece as seguintes contribuições:

a) 1. Proposta de uma Taxonomia Multidimensional: A taxonomia introduzida complementa classificações existentes ao operar em dois eixos:

Eixo 1: Nível de Abstração Cognitiva

- **Computacional:** comportamento entrada-saída; explicações externas.
- **Representacional:** extração de *features* e conceitos; tratamento da superposição.
- **Mecanicista:** engenharia reversa de circuitos e algoritmos internos.

Eixo 2: Epistemologia da Explicação

- **Descritiva:** observação correlacional.
- **Causal:** experimentos que testam hipóteses de causa e efeito.
- **Intervencional:** uso de insights internos para modular o comportamento (e.g., *steering*).
- **Verificável:** mecanismos que permitem checar a fidelidade da explicação.

b) 2. Revisão Crítica do Estado da Arte: Os métodos contemporâneos, incluindo *Sparse Autoencoders* (SAEs), *Induction Heads* [12] e *circuit tracing*, são reclassificados sob a nova taxonomia, permitindo comparações mais estruturadas.

c) 3. Estudo de Caso: A análise da plataforma *Neuronpedia* [13] demonstra como ferramentas modernas de interpretabilidade se encaixam nos níveis representacionais e mecanísticos da taxonomia.

E. Estrutura do Trabalho

Este trabalho está dividido em quatro capítulos:

- **Introdução:** apresenta o problema da caixa-preta, motivação, lacunas e objetivos.
- **Referencial Teórico:** discute definições clássicas e revisões recentes como Zhao et al. [4].
- **Contribuição (Taxonomia e Análise):** descreve a metodologia da Taxonomia Multidimensional e aplica-a ao estado da arte e ao estudo de caso.
- **Conclusão:** sintetiza resultados, limitações e aponta direções futuras para interpretabilidade mecanicista e intervencional.

II. Referencial Teórico

Este capítulo estabelece os fundamentos teóricos essenciais para compreender a interpretabilidade em Grandes Modelos de Linguagem (LLMs). A revisão inicia-se com as definições clássicas da área, propostas por Molnar [5], avança para taxonomias contemporâneas baseadas em arquiteturas Transformer (Zhao et al.) [4] e, em seguida, apresenta um *survey* sobre interpretabilidade interna em redes profundas. Por fim, situa dois eixos emergentes e hoje centrais: a interpretabilidade mecanicista e a engenharia de representações como movimento intervencionista e complementar à linha mecanicista.

A. Fundamentos da Interpretabilidade em Machine Learning

A interpretabilidade, conforme discutido por Molnar [5], não é uma propriedade binária, mas um contínuo de compreensibilidade. Em *Interpretable Machine Learning*, o autor define a interpretabilidade como a capacidade de um humano entender a razão por trás de uma predição. Essa definição desdobra-se em dois tipos de métodos:

- **Intrínsecos (*by design*):** modelos com estrutura transparente, como regressões lineares e árvores de decisão pequenas;
- ***Post-hoc*:** métodos aplicados a modelos já treinados, utilizados para inferir seu comportamento sem alterar sua estrutura.

No entanto, a ascensão de modelos altamente complexos, como LLMs com bilhões de parâmetros, torna inviável o uso de técnicas intrínsecas. Esses modelos operam como "caixas-pretas" e exigem métodos de análise posterior.

A classificação tradicional (*model-agnostic* versus *model-specific*), apesar de historicamente útil, tornou-se insuficiente. LLMs apresentam fenômenos emergentes (como superposição e polissemia neural) que não se ajustam bem a essa dicotomia clássica, motivando a busca por taxonomias mais sofisticadas.

B. Interpretabilidade em Grandes Modelos de Linguagem

Com a adoção generalizada da arquitetura Transformer, a literatura de interpretabilidade entrou em uma nova fase. Zhao et al. [4], no *survey* "Explainability for Large Language Models", propõem uma classificação baseada no paradigma de uso do modelo, e não apenas no acesso aos seus parâmetros.

O *survey* distingue duas grandes categorias:

1) Paradigma de Fine-Tuning Tradicional: Envolve modelos como BERT e RoBERTa, onde o comportamento é moldado via ajustes finos supervisionados. As principais técnicas de interpretabilidade incluem:

- **Feature Attribution:** uso de gradientes, perturbação e *Integrated Gradients*;
- **Probing Classifiers:** sondas que testam se propriedades linguísticas estão codificadas nas camadas internas.

Este paradigma foca em mudanças induzidas pelo ajuste fino e é particularmente útil para tarefas de classificação.

2) Paradigma de Prompting: Refere-se a modelos como GPT e Llama, cujo comportamento é modulado por *prompts* em linguagem natural, sem atualização de pesos. Zhao destaca desafios como:

- Explicar alucinações gerativas;
- Compreender mecanismos do *In-Context Learning*;
- Interpretar cadeias de raciocínio (*Chain-of-Thought*).

Embora essa taxonomia seja valiosa para organizar a literatura sob a ótica do ciclo de vida do modelo, ela ainda opera predominantemente em um nível externalista: informa *quando* explicar, mas não detalha *como* o modelo implementa suas decisões internamente.

C. Interpretabilidade Interna em Redes Neurais Profundas

Apesar de não focar especificamente em LLMs, o *survey* "Toward Transparent AI: A Survey on Interpreting the Inner Structures of Deep Neural Networks" (2022) [11] fornece a base metodológica mais robusta e atual para compreender a interpretabilidade interna (*white-box*) em redes profundas.

Essa revisão sistemática destaca quatro grandes linhas de análise estrutural:

1) **Interpretação de neurônios individuais:** Busca associar um neurônio a um conceito ou função específica. Embora intuitiva, essa abordagem sofre com limitações severas em modelos modernos devido à polissemia neural, onde um único neurônio codifica múltiplos conceitos simultaneamente.
2) **Interpretação de grupos de neurônios (*clusters*):** Técnicas que agrupam dimensões latentes para identificar conceitos distribuídos, reconhecendo que o significado não reside em unidades isoladas.
3) **Análise de camadas e módulos:** Examina o papel funcional de cada bloco da rede (por exemplo, camadas convolucionais, cabeças de atenção ou MLPs).
4) **Métodos baseados em decomposição matemática:** Incluem SVD, PCA, NMF e outros métodos que permitem abstrair representações internas em componentes interpretáveis.

Esse *survey* traz três contribuições fundamentais para o presente trabalho:

(i) Oferece um catálogo sólido de abordagens para interpretar representações internas;
(ii) Demonstra por que analisar "neurônios individuais" é insuficiente em redes modernas;
(iii) Fornece a base conceitual para técnicas de decomposição atualmente usadas na interpretabilidade mecanicista, como *Sparse Autoencoders*.

Assim, *Toward Transparent AI* funciona como uma ponte entre o *Machine Learning* clássico (Molnar) e a engenharia reversa moderna (Elhage et al.).

D. Interpretabilidade Mecanicista em Arquiteturas Transformer

Em paralelo às abordagens estruturais gerais, surge uma linha de pesquisa dedicada a decompor LLMs em circuitos interpretáveis. Essa área, denominada **Interpretabilidade Mecanicista**, foi formalizada por Elhage et al. [12] com *A Mathematical Framework for Transformer Circuits*.

Os autores introduzem dois pilares conceituais:

- **Induction Heads:** Mecanismos emergentes de atenção responsáveis pelo raciocínio de *In-Context Learning*. O surgimento desses mecanismos coincide com uma queda acentuada na função de perda durante o treino, sugerindo uma aquisição algorítmica de capacidades.
- **Decomposição QK/OV:** Separação funcional das matrizes de atenção em dois sub-circuitos:
  - *Query-Key* (QK): circuito que determina *para onde* o modelo deve olhar (cálculo da atenção);
  - *Output-Value* (OV): circuito que transporta a informação selecionada para o fluxo residual.

Esse enquadramento permite a formulação e o teste de hipóteses causais sobre o comportamento dos modelos.

Posteriormente, Elhage et al. [3] apresentam *Toy Models of Superposition*, estudo que demonstra que redes profundas frequentemente representam mais conceitos do que o número de neurônios disponíveis. Esse fenômeno leva à superposição, tornando inútil a análise isolada de neurônios e reforçando a necessidade de métodos estruturais e intervenções no espaço latente.

E. Engenharia de Representação

A Engenharia de Representações (*Representation Engineering*, RepE), proposta por Zou et al. [6], amplia o escopo da interpretabilidade mecanicista ao demonstrar que representações internas não são apenas observáveis, mas também editáveis.

Enquanto a interpretabilidade mecanicista concentra-se na arquitetura e nos pesos estáticos, a RepE atua sobre a dinâmica do modelo:

- Ativações intermediárias;
- Direções conceituais no espaço latente;
- Vetores de *steering* que modulam comportamentos em tempo de inferência.

Zou et al. demonstram que conceitos complexos, como honestidade, utilidade, toxicidade e preferência política, podem ser extraídos e manipulados, permitindo controlar o comportamento do modelo sem a necessidade de alterar seus pesos permanentemente.

F. Síntese do Referencial

A literatura revisada apresenta uma evolução em cinco movimentos complementares:

1) As definições clássicas (Molnar) estruturam a área e introduzem o vocabulário fundamental.
2) As taxonomias de Zhao situam as técnicas no ciclo de vida dos Transformers, embora permaneçam descritivas.
3) O *survey Toward Transparent AI* estabelece a base da interpretabilidade interna moderna, superando a dicotomia *black/white box*.
4) A interpretabilidade mecanicista oferece um arcabouço causal e algorítmico para entender o comportamento específico dos Transformers.
5) A engenharia de representações demonstra que compreender é insuficiente: para garantir segurança, é preciso intervir.

É sobre essa base conceitual, integrando níveis computacionais, representacionais, mecanísticos e intervencionistas, que o próximo capítulo propõe uma nova taxonomia multidimensional para organizar a interpretabilidade em LLMs.

III. Contribuição

A presente abordagem parte das definições clássicas de interpretabilidade propostas por Molnar (2020) [5], que situam os Grandes Modelos de Linguagem (LLMs) como sistemas altamente complexos que demandam abordagens *post-hoc*: técnicas aplicadas após o treinamento visando elucidar seu funcionamento interno. Embora essa distinção inicial entre métodos *Model-Agnostic* (*Black Box*) e *Model-Specific* (*White Box*) ofereça um ponto de partida útil, ela não captura a diversidade, a profundidade analítica e o avanço epistemológico que caracterizam a interpretabilidade moderna em LLMs.

No contexto de LLMs, essa limitação torna-se evidente: técnicas amplamente distintas, como *saliency maps* baseados em gradientes e engenharia reversa de circuitos neurais, são ambas classificadas como *White Box*, apesar de operarem em níveis de abstração, granularidade e finalidade completamente diferentes. Simultaneamente, a taxonomia de Zhao et al. (2024) [4], baseada nos paradigmas de treinamento (*fine-tuning* vs. *prompting*), embora útil para mapear o ciclo de vida do modelo, não oferece discriminação suficiente para comparar técnicas quanto à natureza da explicação, ao escopo cognitivo ou ao seu valor epistemológico.

Dessa forma, torna-se necessário um arcabouço que complemente as classificações existentes, permitindo distinguir entre técnicas descritivas, causais e intervencionistas, assim como entre explicações comportamentais, representacionais e mecanicistas. Esta é precisamente a lacuna que este trabalho busca preencher.

Para organizar o estado da arte de forma coerente, comparável e analiticamente útil, propomos uma **Taxonomia Multidimensional**, estruturada em dois eixos ortogonais, cuja combinação oferece uma visão mais rica das abordagens contemporâneas.

Eixo 1: Nível de Abstração Cognitiva

Este eixo categoriza as técnicas considerando a profundidade da explicação e qual aspecto do sistema o método pretende elucidar. Em vez de agrupar apenas pelo acesso aos parâmetros, este eixo distingue entre níveis de compreensão, do mais superficial ao mais profundo, proporcionando uma hierarquia cognitiva que vai do comportamento observável à engenharia reversa.

Nível Computacional (Input-Output / Comportamental): Analisa o modelo como uma função matemática $f(x) \rightarrow y$, investigando apenas relações entre entradas e saídas. Não precisa ser *Black Box*, mas o foco é o comportamento, não o mecanismo.

- *Exemplos: Chain-of-Thought prompting*, LIME, SHAP, análises de perturbação.

Nível Representacional (Feature-Centric): Investiga o espaço latente e as ativações internas, buscando identificar conceitos e estruturas emergentes, frequentemente contornando o problema da superposição (*polysemanticity*). Foca no que o modelo codifica internamente.

- *Exemplos: Probing classifiers*, *Sparse Autoencoders* (SAEs), *Concept Activation Vectors* (TCAV).

Nível Mecanicista (Algorithmic / Circuit-Centric): O nível mais profundo da explicação, cujo objetivo é realizar engenharia reversa dos circuitos computacionais internos que implementam algoritmos específicos e entender o que o modelo computa.

- *Exemplos: Induction Heads*, *Causal Tracing*, decomposição QK/OV.

Eixo 2: Classificação Epistemológica (Objetivo do Método)

Este eixo categoriza as técnicas com base no objetivo científico da explicação e no tipo de evidência que cada método oferece. Ele formaliza a transição da interpretabilidade como observação para a interpretabilidade como ciência intervencionista.

Descritiva (O que o modelo faz?): Objetivo de descrever fenômenos internos, visualizando padrões de ativação ou regiões de atenção. Não estabelece causalidade (limita-se à correlação).

- *Exemplos:* Visualização de atenção (BertViz), *saliency maps*.

Causal (Por que o modelo faz isso?): Objetivo de demonstrar relações de causa e efeito entre componentes internos e comportamentos do modelo.

- *Exemplos: Ablation studies*, *Causal Tracing*, ROME.

Intervencional (Como manipular o modelo?): Objetivo de controlar o comportamento do modelo através de intervenções diretas nas ativações ou representações.

- *Exemplos: Representation Engineering*, *Activation Addition*, *Steering vectors*.

Verificável (Como verificar se a explicação é válida?): Objetivo de auditoria, medir a fidelidade da explicação, verificar segurança e evitar explicações espúrias.

- *Exemplos:* Métricas de *faithfulness*, estabilidade, detecção de sicofância.

A. Detalhamento do Nível de Abstração Cognitiva

A interpretação de modelos de linguagem pode ocorrer em diferentes "profundidades explicativas". Em vez de tratar as técnicas como simplesmente *White Box* ou *Black Box*, este eixo propõe classificá-las pelo nível cognitivo da explicação, isto é, qual "parte do fenômeno cognitivo" o método tenta descrever.

Essa escolha é motivada por três razões principais:

1) As técnicas modernas não diferem apenas pelo acesso aos parâmetros, mas pelo tipo de pergunta que respondem (ex.: o que o modelo fez? o que ele representou? qual algoritmo ele executou?).
2) LLMs possuem múltiplos níveis de funcionamento, desde o comportamento observável até circuitos internos de atenção; portanto, a taxonomia deve refletir a estrutura da própria arquitetura.
3) Essa divisão aproxima a interpretabilidade de outras ciências cognitivas (como psicologia e neurociência), que também diferenciam explicações comportamentais, representacionais e mecanicistas.

Com isso, o eixo é dividido em três níveis, detalhados a seguir.

1) Nível Computacional (Input–Output / Comportamental): O nível computacional caracteriza técnicas que tratam o modelo como uma função matemática que mapeia entradas em saídas. O foco não está em como o modelo internaliza um conceito, mas em como ele se comporta frente a perturbações ou diferentes estímulos.

Essa categoria é útil quando o objetivo é produzir explicações rápidas, agnósticas ao modelo e com generalização entre arquiteturas. No entanto, ela não investiga representações internas, nem explica mecanismos causais profundos.

Por que esse nível existe? Porque algumas perguntas científicas e aplicadas não exigem conhecimento da estrutura interna, apenas do comportamento externo. Esse tipo de abordagem é comum em *machine learning* clássico e permanece fundamental em LLMs devido à sua escalabilidade e baixo custo.

Uma técnica representativa deste nível é o **In-Context Editing (ICE)** (ou Edição via Contexto), juntamente com estratégias avançadas de *Prompting*. O ICE consiste em modificar o comportamento do modelo fornecendo novos fatos, regras ou demonstrações diretamente na janela de contexto (o *input* x), sem realizar qualquer alteração nos pesos ou acessar as ativações internas da rede neural. A técnica opera sob a premissa de que o modelo é capaz de generalizar ou corrigir sua saída y condicionalmente à nova informação fornecida em x.

Esta abordagem classifica-se no Nível Computacional porque ignora deliberadamente a representação interna (como vetores de ativação) e os mecanismos físicos (como cabeças de atenção). A análise restringe-se à observação da correlação entre a perturbação na entrada e a mudança resultante na distribuição de probabilidade da saída. O modelo é tratado como uma caixa-preta funcional, onde o interesse recai sobre a eficácia da intervenção externa, e não sobre como o algoritmo interno processa essa nova instrução.

Na literatura recente de estado da arte, esta técnica é amplamente utilizada como *baseline* de alto desempenho ou como método de controle comportamental:

1) No *benchmark* **HalluEditBench** [14], o método ICE é avaliado como uma técnica de edição de conhecimento "livre de treinamento" (*training-free*). O estudo demonstra que, para corrigir alucinações, fornecer a resposta correta no contexto pode ser tão ou mais eficaz em métricas de preservação de conhecimento (localidade) do que métodos invasivos que alteram os pesos do modelo (como ROME ou MEMIT), validando a utilidade da análise puramente comportamental.
2) O estudo **AXBENCH** [15] utiliza *prompting* como a linha de base principal para avaliar técnicas de *steering* (pilotagem). Os autores demonstram que, em tarefas de direcionamento de comportamento, simples instruções textuais no nível computacional frequentemente superam métodos complexos de nível representacional, como os *Sparse Autoencoders* (SAEs), evidenciando que a manipulação da entrada continua sendo uma ferramenta poderosa de controle.
3) No trabalho **Disentangling Memory and Reasoning** [16], a técnica é aplicada através da inserção de *tokens* especiais de controle (e.g., `<memory>`, `<reason>`) no *input*. O modelo é treinado para alterar seu comportamento de inferência, separando recuperação de memória de raciocínio lógico, baseando-se apenas nesses sinais computacionais de entrada, sem a necessidade de editar circuitos específicos manualmente.

2) Nível Representacional (Latent / Feature-Centric): O nível representacional agrupa técnicas que investigam o espaço latente, as ativações internas e como conceitos são codificados no modelo. A pergunta aqui não é apenas "o que o modelo faz?", mas "o que o modelo sabe, e como esse conhecimento está distribuído nas camadas internas?". Esse nível cobre o que em neurociência seria chamado de análise de "representações neurais".

Por que esse nível existe? Porque LLMs exibem fenômenos como polissemia neural e superposição, implicando que o entendimento do modelo depende de analisar como conceitos são representados, não apenas como ele se comporta.

A técnica paradigmática deste nível são os **Sparse Autoencoders (SAEs)**. Esta abordagem consiste em treinar uma rede neural autocodificadora esparsa sobre as ativações das camadas internas do LLM. O objetivo é decompor os vetores de ativação densos e opacos, onde múltiplos conceitos estão "superpostos", em uma combinação linear de *features* esparsas e interpretáveis. Diferentemente de uma análise de neurônios individuais (que frequentemente falha devido à polissemia), o SAE identifica direções vetoriais no espaço latente que correspondem a conceitos monosemânticos específicos (como "cibersegurança" ou "estruturas de código").

Esta técnica pertence ao Nível Representacional porque sua unidade de análise não é o comportamento do modelo nem o circuito físico, mas sim a topologia do espaço latente. O foco recai sobre a decodificação da "linguagem interna" do modelo, isolando os conceitos que o modelo "sabe" e está manipulando em um dado momento, independentemente de como esse conhecimento será usado para gerar a próxima palavra. É uma tentativa direta de mapear o território semântico interno da rede neural.

Na literatura atual de estado da arte, os SAEs são aplicados tanto para interpretação quanto para intervenções cirúrgicas:

1) No artigo "*Model Unlearning via Sparse Autoencoder Subspace Guided Projections*" (SSPU) [17], a técnica é utilizada para segurança e privacidade. Os autores utilizam SAEs para identificar *features* latentes associadas a conhecimentos perigosos (ex: bioterrorismo ou ciberataques). Em vez de apenas observar essas *features*, eles constroem um subespaço ortogonal a elas e projetam os pesos do modelo para "esquecer" esses conceitos específicos sem degradar o conhecimento geral do modelo.
2) O trabalho "*AXBENCH: Steering LLMs?*" [15] avalia os SAEs em uma tarefa de *steering*. O estudo investiga se a identificação de *features* latentes via SAEs permite controlar a geração do modelo. Embora os autores descubram que métodos supervisionados simples (como *Difference-in-Means*) podem superar SAEs em eficácia de pilotagem, o uso de SAEs permanece central para a detecção de conceitos não supervisionados, servindo como uma "sonda" para verificar o que está ativo no espaço representacional durante a inferência.
3) No contexto de "*Locate-then-Merge*" [18], a análise no nível representacional é usada para mitigar o esquecimento catastrófico em modelos multimodais. O estudo analisa como a mudança nos parâmetros de neurônios específicos (vistos como *features* na base canônica) correlaciona-se com a aquisição de habilidades visuais versus a perda de habilidades linguísticas, propondo fusões seletivas baseadas na magnitude dessas representações internas.

3) Nível Mecanicista (Circuit-Based / Algorithmic): O nível mecanicista é o mais profundo da taxonomia. Ele não apenas descreve representações, mas busca reconstituir o algoritmo interno que o modelo está executando: uma espécie de "engenharia reversa funcional". A pergunta típica desse nível é: "qual circuito implementa esta tarefa e como ele opera?".

Por que esse nível existe? Porque LLMs realizam comportamentos emergentes que só podem ser explicados por interações específicas entre cabeças de atenção, neurônios MLP e fluxos de informação, o que requer decompor a arquitetura em blocos computacionais causais. Essa linha fundamenta a chamada Interpretabilidade Mecanicista, liderada por trabalhos da Anthropic e grupos independentes.

A técnica paradigmática deste nível é a **Descoberta de Circuitos** (*Circuit Discovery*), frequentemente operacionalizada por métodos como *Edge Attribution Patching* (EAP) ou *Causal Tracing*. Essas abordagens buscam isolar o subgrafo computacional mínimo dentro da rede, composto por cabeças de atenção específicas, neurônios MLP e suas conexões (arestas), que é causalmente responsável por executar uma tarefa, como resolver uma equação matemática ou recuperar um fato histórico. Em vez de analisar o modelo inteiro, isola-se o "programa" esparso que ele executa para aquele *input*.

Esta abordagem classifica-se no Nível Mecanicista porque transcende a identificação de conceitos estáticos (foco do nível representacional) para mapear a dinâmica do processamento de informação. A unidade de análise aqui é a interação causal: como a saída de uma cabeça de atenção na camada L_i é escrita no fluxo residual e subsequentemente lida e transformada por um MLP na camada L_j. O objetivo final é reduzir a complexidade da rede neural a um algoritmo legível por humanos, detalhando o passo a passo das operações internas que levam à resposta.

Na literatura de estado da arte, essas técnicas são fundamentais para diagnósticos profundos e intervenções arquiteturais:

1) No artigo "*Towards Understanding Fine-Tuning Mechanisms of LLMs via Circuit Analysis*" [19], a técnica *Edge Attribution Patching with Integrated Gradients* (EAP-IG) é aplicada para mapear circuitos de raciocínio matemático. O estudo revela um mecanismo interno surpreendente: o *fine-tuning* não funciona apenas ajustando nós (neurônios) existentes, mas principalmente reestruturando as arestas do grafo computacional. Isso demonstra que o modelo aprende novas tarefas "recabeando" o fluxo de informação entre seus componentes.
2) O trabalho "*Back Attention: Understanding and Enhancing Multi-Hop Reasoning*" [20] utiliza uma análise mecanicista via *Logit Flow* para decompor o algoritmo de recuperação de conhecimento. Os autores conseguem segmentar o processo de inferência em quatro etapas algorítmicas distintas e propõem o mecanismo *Back Attention*, que permite a camadas inferiores acessarem estados futuros para corrigir falhas no circuito de raciocínio.
3) Em "*Taming Knowledge Conflicts in Language Models*" [21], a análise mecanicista é usada para identificar componentes funcionais específicos, denominados "cabeças de memória" e "cabeças de contexto". Ao realizar intervenções causais (*knocking out*) nesses módulos, os autores descobrem que o modelo processa memória paramétrica e contexto em superposição. Esse *insight*

mecanicista fundamenta o método JUICE, que intervém na saída dessas cabeças específicas para controlar a priorização do conhecimento.

B. Detalhamento do Nível Epistemológico

Enquanto o Eixo 1 organiza as técnicas pela profundidade cognitiva da explicação, o Eixo 2 propõe uma classificação baseada na natureza epistemológica do conhecimento produzido por cada método: o tipo de evidência que a explicação fornece e qual pergunta científica ela responde.

Essa abordagem parte da premissa de que entender o que um modelo faz, por que faz, como intervir e como verificar a fidelidade da explicação constituem etapas epistemológicas distintas, todas fundamentais para a interpretabilidade madura de LLMs.

A motivação central desse eixo é que duas técnicas podem acessar o mesmo nível cognitivo (ex.: representacional), mas com objetivos epistêmicos radicalmente diferentes. Por exemplo, *Sparse Autoencoders* e *Activation Steering* lidam com representações internas, mas o primeiro busca descrever conceitos, enquanto o segundo busca manipular o comportamento do modelo. Assim, a classificação epistemológica permite entender o propósito da explicação, complementando o eixo cognitivo.

O eixo é dividido em quatro categorias:

1) Explicações Descritivas (Observacionais): **Definição:** Métodos que buscam observar o comportamento do modelo sem estabelecer causalidade entre componentes internos e resultados.

- *Pergunta epistemológica associada:* "O que o modelo está fazendo?"

Essas técnicas fornecem evidências correlacionais, como visualizações, ativações e padrões de atenção. São úteis para diagnóstico inicial, mas não permitem concluir se o fenômeno observado é causal ou apenas um artefato estatístico.

Por que este nível existe? Porque a maior parte da XAI clássica foi construída como uma ciência observacional: observar padrões, destacar regiões importantes, visualizar pesos. Esses métodos continuam essenciais como ferramentas de triagem, mesmo que não respondam perguntas causais.

Uma técnica paradigmática deste nível é o **Logit Lens** (e sua variante *Logit Flow*). Esta abordagem projeta os estados latentes das camadas intermediárias do modelo diretamente no vocabulário de saída, permitindo "ler" o que o modelo está prevendo em cada etapa do processamento, antes da camada final. Esta técnica classifica-se como Descritiva porque não altera o funcionamento do modelo nem prova que aquela representação intermediária é a causa única da saída final; ela apenas traduz a ativação interna em um formato legível (tokens), oferecendo um diagnóstico visual da evolução da predição.

Na literatura recente, esta técnica é essencial para diagnósticos de processamento de informação:

1) No artigo "*Lost in Multilinguality*" [22], o *Logit Lens* é a ferramenta primária para descrever o fenômeno da "transição de língua". Os autores observam que, nas camadas médias, o modelo processa a resposta correta em um "espaço de conceito" (frequentemente em inglês), e apenas nas camadas finais tenta traduzir para a língua alvo. A falha observada nessa tradução explica a inconsistência factual entre línguas.
2) O trabalho "*Back Attention*" [20] utiliza o *Logit Flow* para descrever as quatro etapas sequenciais do raciocínio em *transformers* (enriquecimento de sujeito, extração de atributo, etc.). A técnica permite visualizar como a probabilidade (*logit*) da resposta correta flutua camada a camada, diagnosticando onde o raciocínio falha em tarefas de múltiplos saltos.

Outro exemplo relevante é a **Projeção de Espaço de Ativação** (via MDS/PCA). No estudo "*Shaping the Safety Boundaries*" [23], ela é usada para descrever topologicamente a diferença entre *prompts* benignos, prejudiciais e de *jailbreak*. A visualização revela que ataques de *jailbreak* deslocam as ativações para fora de uma "fronteira de segurança" latente. É uma análise puramente observacional que fundamenta a defesa proposta posteriormente.

2) Explicações Causais (Explicativas): **Definição:** Métodos que investigam relações de causa e efeito, buscando provar que um componente interno específico é responsável por um comportamento.

- *Pergunta epistemológica associada:* "Por que o modelo faz isso?" (Se eu remover X, o comportamento Y muda?).

Diferentemente dos métodos descritivos, que mostram correlação, métodos causais utilizam intervenções controladas (como ablação ou isolamento) para estabelecer a necessidade ou suficiência de um componente.

Por que este nível existe? Porque entender correlação não basta quando o objetivo é garantir segurança, robustez ou auditoria. A causalidade permite distinguir explicações verdadeiras de explicações plausíveis porém falsas.

Técnica Representativa: *Causal Tracing* e Ablação (*Knocking Out*). O *Causal Tracing* introduz ruído em partes do modelo para ver se a saída muda, restaurando posteriormente ativações específicas para ver se a saída é recuperada. Já a ablação zera ou remove a contribuição de componentes específicos (como cabeças de atenção) para testar sua função.

Evidência na Literatura:

1) No artigo "*Taming Knowledge Conflicts*" [21], a técnica de *Knocking Out* é usada para estabelecer causalidade sobre "cabeças de memória" e "cabeças de contexto". Ao remover cabeças específicas e observar a queda na probabilidade da resposta paramétrica versus contextual, os autores provam causalmente que certas cabeças não são puramente dedicadas a uma função, mas operam em superposição.
2) O benchmark **HalluEditBench** [14] discute métodos como ROME (*Rank-One Model Editing*), que se baseiam em *Causal Tracing* para localizar o local exato onde uma memória factual está armazenada (geralmente em camadas MLP intermediárias) antes de editá-la. A premissa é que a edição só funciona porque a relação causal foi previamente estabelecida.

3) Explicações Intervencionais (Controle / Steering): **Definição:** Métodos que utilizam o conhecimento causal ou representacional para controlar ativamente o comportamento do modelo em tempo de inferência, sem necessariamente retreiná-lo.

- *Pergunta epistemológica associada:* "Como podemos manipular o modelo para atingir um objetivo específico?"

Este nível representa a transição da ciência passiva para a engenharia ativa. O objetivo não é mais apenas entender o erro, mas intervir na computação para corrigi-lo ou direcioná-lo.

Técnica Representativa: *Activation Steering* e *Feature Clamping*. Esta técnica consiste em adicionar um "vetor de direção" (*steering vector*) às ativações internas do modelo durante a inferência, ou fixar (*clamp*) o valor de certas *features* (descobertas por exemplo via SAEs) para forçar o modelo a adotar um comportamento (ex: recusar conteúdo tóxico ou focar em um tópico).

Evidência na Literatura:

1) O trabalho "**AXBENCH**" [15] é um estudo extensivo sobre intervenção. Ele compara métodos como ReFT (*Representation Finetuning*) e *Steering* com *Sparse Autoencoders* (SAEs). O estudo demonstra que adicionar vetores calculados via SAEs ou métodos supervisionados (como *Difference-in-Means*) pode controlar se o modelo responde a uma instrução ou menciona um conceito específico, validando a intervenção como ferramenta de controle de segurança.
2) No artigo "*Model Unlearning via Sparse Autoencoder Subspace Guided Projections*" (SSPU) [17], a intervenção é levada ao extremo do "desaprendizado". O método identifica um subespaço de *features* perigosas (via SAE) e projeta os pesos do modelo para um subespaço ortogonal, efetivamente removendo a capacidade do modelo de processar aquele conceito (intervenção cirúrgica nos pesos guiada por representações).
3) A técnica **JUICE**, proposta em "*Taming Knowledge Conflicts*" [21], é uma intervenção em tempo de teste que executa o modelo duas vezes: a primeira para identificar o estado das cabeças de atenção e a segunda para intervir nelas, amplificando a memória ou o contexto conforme desejado.

4) Explicações Verificáveis (Auditoria / Avaliação de Fidelidade): **Definição:** Métodos focados em avaliar se uma explicação ou edição é verdadeira, robusta e não causa efeitos colaterais indesejados.

- *Pergunta epistemológica associada:* "A explicação ou intervenção é fiel e segura?"

Este nível é crítico para garantir que as técnicas dos níveis anteriores não sejam "alucinações explicativas". Ele foca em métricas de fidelidade, especificidade e generalização.

Técnica Representativa: Métricas de Localidade e Portabilidade (*Evaluation Frameworks*). Não se trata de um algoritmo de visualização, mas de protocolos de teste que verificam se uma mudança interna teve o efeito esperado externamente (eficácia) sem quebrar outras partes do modelo (localidade).

Evidência na Literatura:

1) O trabalho "*The Mirage of Model Editing*" [24] é uma crítica direta à falta de verificabilidade real na área. Os autores demonstram que métodos considerados eficazes em avaliações sintéticas falham em cenários reais. Este nível epistemológico é representado aqui pela proposta de métricas que verificam a robustez da edição sob geração livre, em vez de *teacher forcing*, revelando que a "fidelidade" reportada anteriormente era inflada.
2) No **HalluEditBench** [14], a verificabilidade é sistematizada em cinco dimensões (Eficácia, Generalização, Portabilidade, Localidade e Robustez). O estudo mostra, por exemplo, que métodos de edição podem ter alta eficácia (corrigem o fato), mas baixa portabilidade (o modelo não consegue usar o novo fato para responder perguntas derivadas), expondo a fragilidade da intervenção.

C. Estudo de Caso: Neuronpedia e Attribution Graph

1) O que é a Neuronpedia e qual é sua relevância?: A **Neuronpedia** [13] é uma plataforma aberta de interpretabilidade desenvolvida em colaboração por equipes de pesquisa da Anthropic, Google DeepMind e outras instituições. Seu propósito principal é permitir que pesquisadores e usuários explorem, analisem e manipulem representações internas de modelos de linguagem em larga escala. A plataforma integra ferramentas avançadas de inspeção, engenharia de representações e *steering*, constituindo atualmente um dos ambientes mais completos para experimentos de interpretabilidade em LLMs.

A Figura 1 ilustra a interface da plataforma ao analisar um neurônio específico, destacando os *tokens* que mais o ativam e as predições associadas.

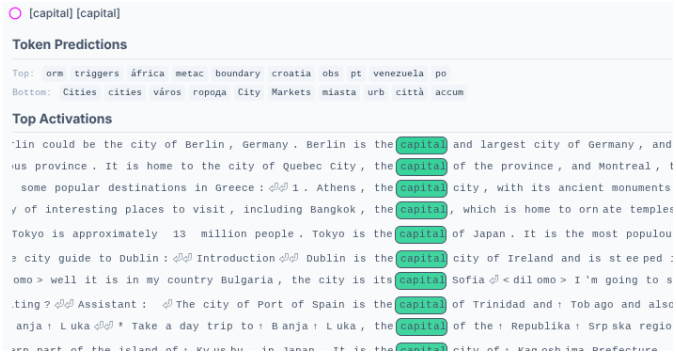

Fig. 1. Visualização de ativação de um neurônio na Neuronpedia. O painel exibe os tokens do texto que ativam o neurônio (destacados em verde) e as predições de tokens subsequentes.

Entre suas funcionalidades centrais, destacam-se:

- **Circuit Tracer:** Permite visualizar e rastrear as etapas internas de raciocínio do modelo a partir de um *prompt*, seguindo a linha dos trabalhos de *circuit tracing* introduzidos por Ameisen et al. [25] e Lindsey et al.

[26]. Essa ferramenta reconstrói caminhos algorítmicos aproximados dentro do modelo.

- **Explore:** Oferece uma interface interativa para navegar por milhões de ativações, conceitos, vetores, latentes, explicações geradas automaticamente e metadados. A ferramenta inclui suporte a *probes*, *transcoders*, *features* aprendidas e vetores personalizados.
- **Steer:** Permite modificar o comportamento do modelo intervindo diretamente nas ativações internas por meio de *steering vectors*, *features* ou conceitos aprendidos. Essa funcionalidade pode ser aplicada tanto em modelos de instrução quanto em modelos voltados para raciocínio.
- **Search:** Possibilita realizar buscas semânticas no espaço de *features*, seja por similaridade textual ou por inferência direta em modelos hospedados na plataforma.

Essas capacidades tornam a Neuronpedia um ambiente essencial para a investigação prática de técnicas contemporâneas de interpretabilidade, especialmente aquelas localizadas no nível representacional e mecanicista da taxonomia proposta neste trabalho.

2) O que é um Attribution Graph?: Os *Attribution Graphs* constituem uma técnica de interpretabilidade destinada a decompor e visualizar o fluxo de informação que ocorre internamente quando um modelo processa um determinado *input*. O objetivo é reconstruir, em forma gráfica, a "cadeia de raciocínio" computacional do modelo: quais conceitos foram ativados, como eles interagem e de que maneira contribuíram para a predição final. A Figura 2 apresenta um exemplo visual dessa estrutura.

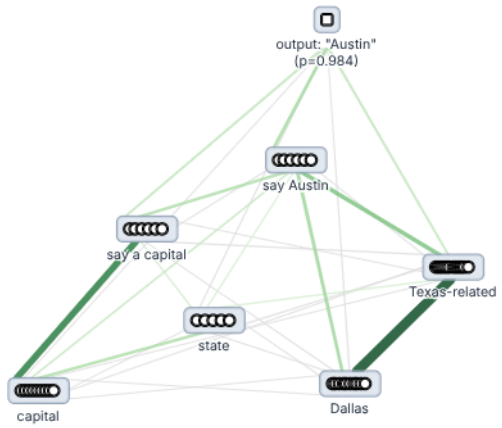

Fig. 2. Exemplo de um Attribution Graph. O grafo mapeia a relação causal entre os tokens de entrada (ex: "Dallas"), features intermediárias ativadas (ex: "Texas-related", "state") e a saída final do modelo ("Austin").

Um *attribution graph* é composto por dois elementos fundamentais:

A. Nós (Nodes) Cada nó representa um componente da computação interna do modelo. Entre os tipos principais:

- **Embeddings (Nível Inferior):** Representados como quadrados na parte inferior do gráfico. Correspondem aos *tokens* do *prompt*. Exemplo: em "*the capital of the state containing Dallas is*", *tokens* como *capital*, *state* e *Dallas* aparecem como nós de entrada.
- **Features (Nível Intermediário):** Representadas como círculos ou losangos. São conceitos latentes aprendidos pelo modelo ao longo do treinamento. Frequentemente são descobertas por meio de *Transcoders* (como *Cross-Layer Transcoders*), que substituem as MLPs tradicionais por componentes mais interpretáveis. Uma *feature* corresponde a uma dimensão ativa no espaço latente do modelo e pode ser associada a exemplos do conjunto de dados que a ativam fortemente, indicar *tokens* que tende a "votar" para que o modelo gere, ou revelar conceitos como localizações, entidades, emoções, sintaxe etc.
- **Nós de Erro (ERR-MLP):** Representam a parte da computação que o *transcoder* não conseguiu explicar, indicando informação residual ainda não compreendida.
- **Logits de Saída (Nível Superior):** Representados como quadrados no topo do gráfico. Correspondem à predição final do modelo.

B. Arestas (Edges) As arestas representam relações de influência entre os nós:

- Interpretam como a mudança em um nó afeta nós posteriores, modelando uma forma aproximada de atribuição causal linear.
- A maior parte das arestas irrelevantes é podada, para deixar visíveis apenas os caminhos mais influentes.
- O gráfico conta uma "história computacional", revelando como conceitos intermediários se combinam para produzir a saída final.

Há, porém, limitações: *attribution graphs* enfatizam relações positivas (por que o modelo disse X), mas geralmente não mostram relações inibitórias (por que o modelo não disse Y), que também são importantes para a completude causal.

Classificação do Attribution Graph na Taxonomia Proposta:

- **Eixo 1 - Nível de Abstração Cognitiva: Representacional**, com elementos **mecanicistas**. O *attribution graph* opera primariamente no espaço latente (representações e *features*), mas também tenta reconstruir caminhos computacionais consistentes com circuitos internos. Portanto, posiciona-se principalmente no Nível Representacional, com extensão parcial ao Nível Mecanicista, quando a técnica tenta aproximar "circuitos" funcionais.
- **Eixo 2 - Epistemologia da Explicação: Causal**, com caráter **intervencional** quando usada em conjunto com *steering*. O método busca explicar por que o modelo chegou à predição observada, mapeando relações de causa e efeito entre ativações internas. Quando usado para testar o efeito de manipular *features*, aproxima-se do nível intervencional.

3) Exemplo de Análise: Capital do Estado que Contém Dallas: Para ilustrar a aplicabilidade do *attribution graph*, analisamos o caso descrito pela plataforma Neuronpedia (visualizado na Figura 2) utilizando o modelo Gemma 2B com o *prompt*: "*fact the capital of the state containing Dallas is*"

O modelo produz corretamente o *token* "Austin". O *attribution graph* permite decompor essa predição em etapas interpretáveis:

1) **Ativação Inicial dos Embeddings:** O *token* "Dallas" ativa fortemente *features* associadas ao estado do Texas.
2) **Ativação de Features Intermediárias:** O gráfico identifica *features* relacionadas à geografia texana e outras relacionadas ao conceito de "capital".
3) **Composição Conceitual:** A saída "Austin" emerge da interseção entre *features* representando "estado do Texas" e *features* representando "ser uma capital".
4) **Mecanismo Aditivo:** A predição final surge da combinação linear das duas direções conceituais no espaço latente: direção Texas e direção capital.
5) **Experimento de Steering:** Ao aplicar *steering vectors* negativos sobre as *features* de Texas, o modelo deixa de prever "Austin", confirmando o papel causal dessas representações na decisão final.

Esse exemplo demonstra como o *attribution graph* integra explicação representacional, causalidade e intervenção, alinhando-se diretamente com a taxonomia proposta neste trabalho.

IV. Conclusão e Trabalhos Futuros

A opacidade dos Grandes Modelos de Linguagem (LLMs) representa um dos desafios técnicos e éticos mais prementes da inteligência artificial contemporânea. À medida que modelos como GPT-4 e Gemini 3 permeiam infraestruturas críticas, a incapacidade de explicar *como* e *por que* uma determinada saída foi gerada deixa de ser uma questão puramente acadêmica para tornar-se um requisito de segurança. Este trabalho buscou mitigar esse problema não pela proposição de um novo algoritmo de visualização, mas pela estruturação do campo através de uma nova lente analítica: a **Taxonomia Multidimensional**.

A revisão crítica da literatura demonstrou que as classificações tradicionais, sejam baseadas no acesso aos parâmetros (*Black-Box* vs. *White-Box*) ou no paradigma de treinamento (*Fine-tuning* vs. *Prompting*), tornaram-se insuficientes. Elas falham em capturar a nuance de técnicas modernas que, embora acessem os pesos do modelo ("White-Box"), operam em níveis de abstração radicalmente diferentes: desde a simples correlação de ativações até a engenharia reversa de circuitos algorítmicos.

A principal contribuição deste estudo foi a formalização de dois eixos ortogonais para a classificação de técnicas de interpretabilidade: o **Nível de Abstração Cognitiva** (Computacional, Representacional, Mecanicista) e o **Propósito Epistêmico** (Descritivo, Causal, Intervencional, Verificável). A aplicação desta taxonomia ao estado da arte permitiu reavaliar métodos emergentes com maior precisão. Demonstramos, por exemplo, que os *Sparse Autoencoders* (SAEs) não são apenas ferramentas de visualização, mas instrumentos de intervenção cirúrgica no nível representacional, capazes de mitigar o fenômeno da polissemia neural. Da mesma forma, a análise via *Attribution Graphs* na plataforma Neuronpedia ilustrou como a explicação pode transitar do nível descritivo para o causal, isolando sub-redes responsáveis por comportamentos específicos.

Em suma, este trabalho evidencia uma transição paradigmática na área: a interpretabilidade está evoluindo de uma ciência passiva e observacional para uma ciência ativa e intervencionista. Não basta mais perguntar "o que o modelo está olhando?"; a fronteira do conhecimento reside agora em perguntar "como podemos editar esse conceito interno para alterar o comportamento do modelo de forma previsível?".

A. Trabalhos Futuros

Apesar dos avanços sistematizados neste trabalho, a área de interpretabilidade em LLMs permanece em estágio pré-paradigmático em muitos aspectos. Com base nas limitações identificadas e nas tendências emergentes, sugerimos as seguintes direções para pesquisas futuras:

- **Padronização de Métricas de Fidelidade (*Faithfulness*):** Um dos maiores obstáculos identificados é a ausência de *ground truth* para explicações. Trabalhos futuros devem focar no desenvolvimento de *benchmarks* rigorosos que penalizem explicações que soam plausíveis para humanos, mas que não refletem o verdadeiro mecanismo causal do modelo (o problema da "alucinação explicativa").
- **Escalabilidade da Interpretabilidade Mecanicista:** Técnicas como *Circuit Tracing* e *Induction Heads* foram validadas predominantemente em modelos de pequeno e médio porte. Investigar se esses circuitos persistem, desaparecem ou se transformam em modelos na escala de trilhões de parâmetros é uma questão aberta crucial. A automação da descoberta de circuitos é um passo necessário para lidar com essa complexidade.
- **Interpretabilidade Multimodal:** Com a ascensão de modelos que processam texto, imagem e áudio simultaneamente (como o Gemini 3), surge a necessidade de estender a Taxonomia Multidimensional para cobrir representações multimodais. Como a "superposição" ocorre quando conceitos visuais e textuais compartilham o mesmo espaço latente?
- **Steering Vetorial para Alinhamento de Segurança:** Aprofundar o uso de *Representation Engineering* como alternativa ao *Reinforcement Learning from Human Feedback* (RLHF). Enquanto o RLHF atua no comportamento externo, intervenções diretas no espaço latente (via *steering vectors*) oferecem um caminho promissor para impedir a geração de conteúdo tóxico ou alucinado "antes" que ele seja decodificado.

A interpretabilidade não é um destino, mas um processo contínuo de tradução entre a matemática de alta dimensão e a cognição humana. Espera-se que a estrutura proposta neste trabalho sirva como um mapa útil para navegar esse território em rápida expansão.

References

[1] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat,

R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, and B. Zoph, "GPT-4 Technical Report," OpenAI, Tech. Rep., 03 2023. [Online]. Available: https://arxiv.org/abs/2303.08774

[2] Google, "Gemini 3," Google The Keyword Blog, 2025, acessado em: 24 nov. 2025. [Online]. Available: https://blog.google/products/gemini/gemini-3/

[3] N. Elhage, T. Hume, C. Olsson, N. Schiefer, T. Henighan, S. Kravec, Z. Hatfield-Dodds, R. Lasenby, D. Drain, C. Chen, R. Grosse, S. McCandlish, J. Kaplan, D. Amodei, M. Wattenberg, and C. Olah, "Toy Models of Superposition," *Transformer Circuits Thread*, 2022. [Online]. Available: https://transformer-circuits.pub/2022/toy_model/index.html

[4] H. Zhao, H. Chen, F. Yang, N. Liu, H. Deng, H. Cai, S. Wang, D. Yin, and M. Du, "Explainability for Large Language Models: A Survey," *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 2, 2 2024. [Online]. Available: https://doi.org/10.1145/3639372

[5] C. Molnar, *Interpretable Machine Learning*, 3rd ed. Christoph Molnar, 2025. [Online]. Available: https://christophm.github.io/interpretable-ml-book

[6] A. Zou, L. Phan, S. Chen, J. Campbell, P. Guo, R. Ren, A. Pan, X. Yin, M. Mazeika, A.-K. Dombrowski, S. Goel, N. Li, M. J. Byun, Z. Wang, A. Mallen, S. Basart, S. Koyejo, D. Song, M. Fredrikson, J. Z. Kolter, and D. Hendrycks, "Representation Engineering: A Top-Down Approach to AI Transparency," 2023. [Online]. Available: https://arxiv.org/abs/2310.01405

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 6000–6010.

[8] A. Radford and K. Narasimhan, "Improving Language Understanding by Generative Pre-Training," 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:49313245

[9] Z. C. Lipton, "The Mythos of Model Interpretability," *Communications of the ACM*, vol. 61, no. 10, pp. 36–43, 9 2018. [Online]. Available: https://doi.org/10.1145/3233231

[10] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[11] T. Rauker, A. Ho, S. Casper, and D. Hadfield-Menell, "Toward Transparent AI: A Survey on Interpreting the Inner Structures of Deep Neural Networks," in *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2023, pp. 464–483.

[12] N. Elhage, N. Nanda, C. Olsson, T. Henighan, N. Joseph, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, N. DasSarma, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah, "A Mathematical Framework for Transformer Circuits," *Transformer Circuits Thread*, 2021. [Online]. Available: https://transformer-circuits.pub/2021/framework/index.html

[13] Neuronpedia Team, "Neuronpedia: An Open Platform for Interpretability Research," 2025, acessado em: 24 nov. 2025. [Online]. Available: https://www.neuronpedia.org/

[14] B. Huang, C. Chen, X. Xu, A. Payani, and K. Shu, "Can knowledge editing really correct hallucinations?" in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=hmDt068MoZ

[15] Z. Wu, A. Arora, A. Geiger, Z. Wang, J. Huang, D. Jurafsky, C. D. Manning, and C. Potts, "AxBench: Steering LLMs? Even Simple Baselines Outperform Sparse Autoencoders," in *Forty-second International Conference on Machine Learning*, 2025. [Online]. Available: https://openreview.net/forum?id=K2CckZjNy0

[16] M. Jin, W. Luo, S. Cheng, X. Wang, W. Hua, R. Tang, W. Y. Wang, and Y. Zhang, "Disentangling memory and reasoning ability in large language models," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Vienna, Austria: Association for Computational Linguistics, Jul. 2025, pp. 1681–1701. [Online]. Available: https://aclanthology.org/2025.acl-long.84/

[17] X. Wang, Z. Li, B. Wang, Y. Hu, and D. Zou, "Model unlearning via sparse autoencoder subspace guided projections," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds. Suzhou, China: Association for Computational Linguistics, Nov. 2025, pp. 26 541–26 557. [Online]. Available: https://aclanthology.org/2025.emnlp-main.1348/

[18] Z. Yu and S. Ananiadou, "Locate-then-merge: Neuron-level parameter fusion for mitigating catastrophic forgetting in multimodal LLMs," in *Findings of the Association for Computational Linguistics: EMNLP 2025*, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds. Suzhou, China: Association for Computational Linguistics, Nov. 2025, pp. 7065–7078. [Online]. Available: https://aclanthology.org/2025.findings-emnlp.372/

[19] X. Wang, Y. Hu, W. Du, R. Cheng, B. Wang, and D. Zou, "Towards Understanding Fine-Tuning Mechanisms of LLMs via Circuit Analysis," in *Forty-second International Conference on Machine Learning*, 2025. [Online]. Available: https://openreview.net/forum?id=45EIiFd6Oa

[20] Z. Yu, Y. Belinkov, and S. Ananiadou, "Back attention: Understanding and enhancing multi-hop reasoning in large language models," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds. Suzhou, China: Association for Computational Linguistics, Nov. 2025, pp. 11 268–11 283. [Online]. Available: https://aclanthology.org/2025.emnlp-main.567/

[21] G. Li, Y. Chen, and H. Tong, "Taming knowledge conflicts in language models," in *Forty-second International Conference on Machine Learning*, 2025. [Online]. Available: https://openreview.net/forum?id=0cEZyhHEks

[22] M. Wang, H. Adel, L. Lange, Y. Liu, E. Nie, J. Strötgen, and H. Schuetze, "Lost in multilinguality: Dissecting cross-lingual factual inconsistency in transformer language models," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Vienna, Austria: Association for Computational Linguistics, Jul. 2025, pp. 5075–5094. [Online]. Available: https://aclanthology.org/2025.acl-long.253/

[23] L. Gao, J. Geng, X. Zhang, P. Nakov, and X. Chen, "Shaping the safety boundaries: Understanding and defending against jailbreaks in large language models," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Vienna, Austria: Association for Computational Linguistics, Jul. 2025, pp. 25 378–25 398. [Online]. Available: https://aclanthology.org/2025.acl-long.1233/

[24] Anonymous, "The mirage of model editing: Revisiting evaluation in the wild," in *Submitted to ACL Rolling Review - February 2025*, 2025, under review. [Online]. Available: https://openreview.net/forum?id=SPScLo90NR

[25] E. Ameisen, J. Lindsey, A. Pearce, W. Gurnee, N. L. Turner, B. Chen, C. Citro, D. Abrahams, S. Carter, B. Hosmer, J. Marcus, M. Sklar, A. Templeton, T. Bricken, C. McDougall, H. Cunningham, T. Henighan, A. Jermyn, A. Jones, A. Persic, Z. Qi, T. Ben Thompson, S. Zimmerman, K. Rivoire, T. Conerly, C. Olah, and J. Batson, "Circuit Tracing: Revealing Computational Graphs in Language Models," *Transformer Circuits Thread*, 2025. [Online]. Available: https://transformer-circuits.pub/2025/attribution-graphs/methods.html

[26] J. Lindsey, W. Gurnee, E. Ameisen, B. Chen, A. Pearce, N. L. Turner, C. Citro, D. Abrahams, S. Carter, B. Hosmer, J. Marcus, M. Sklar, A. Templeton, T. Bricken, C. McDougall, H. Cunningham, T. Henighan, A. Jermyn, A. Jones, A. Persic, Z. Qi, T. B. Thompson, S. Zimmerman, K. Rivoire, T. Conerly, C. Olah, and J. Batson, "On the Biology of a Large Language Model," *Transformer Circuits Thread*, 2025. [Online]. Available: https://transformer-circuits.pub/2025/attribution-graphs/biology.html