CORRECTING INFLUENCE: UNBOXING LLM OUTPUTS WITH ORTHOGONAL LATENT SPACES

Anonymous authors
Paper under double-blind review

ABSTRACT

A critical step for reliable large language models (LLMs) use in healthcare is to attribute predictions to their training data, akin to a medical case study. This requires token-level precision: pinpointing not just which training examples influence a decision, but which tokens within them are responsible. While *influence functions* offer a principled framework for this, prior work is restricted to *autoregressive* settings and relies on an implicit assumption of *token independence*, rendering their identified influences unreliable. We introduce a flexible framework that infers *token-level influence* through a latent mediation approach for *general prediction tasks*. Our method attaches *sparse autoencoders* to any layer of a pretrained LLM to learn a basis of approximately independent latent features. Unlike prior methods where influence decomposes additively across tokens, influence computed over latent features is inherently *non-decomposable*. To address this, we introduce a novel method using *Jacobian-vector products*. Token-level influence is obtained by propagating latent attributions back to the input space via token activation patterns. We scale our approach using efficient inverse-Hessian approximations. Experiments on medical benchmarks show our approach identifies sparse, interpretable sets of tokens that *jointly* influence predictions. Our framework enhances trust and enables model auditing, generalizing to any high-stakes domain requiring transparent and accountable decisions.

1 INTRODUCTION

The deployment of LLMs in high-stakes domains like healthcare hinges on a critical and unmet requirement: the ability to audit a model's reasoning by tracing its predictions directly to the evidence in its training data. This need for verifiability is urgent, as LLMs are increasingly explored for clinical tasks such as diagnostic support and treatment planning, where errors can have severe consequences (Singhal et al., 2023; Topol, 2019). Without this capability—akin to a clinician demanding the source for a medical decision—LLMs remain unverifiable black boxes. Their tendency to hallucinate (Ji et al., 2023) and their susceptibility to spurious correlations present in training data (Oberst & Sontag, 2019) pose significant safety risks, undermining the trust required for clinical adoption (Futoma et al., 2020; Ghassemi et al., 2021).

This fundamental need for evidence-based reasoning is not adequately addressed by prevailing interpretability methods. Techniques like Chain-of-Thought prompting generate rationales that are often post hoc justifications rather than faithful reflections of the model's true decision process (Turpin et al., 2023; Barez et al., 2025). Other popular approaches, such as attention visualization (Wiegreffe & Pinter, 2019; Jain & Wallace, 2019) or gradient-based feature attribution (Sundararajan et al., 2017a), are limited to explaining a single forward pass of a model. They operate within the context of a given input, providing no insight into how prior training experiences shaped the model's fundamental behavioral patterns and knowledge (Feldman & Zhang, 2020). This represents a critical limitation for clinical deployment, where the ability to pinpoint the exact training evidence behind a prediction—not just generate plausible-sounding rationales—is essential for medical professionals to validate the model's logic against established knowledge, fact-check its conclusions, and ultimately build the trust required for adoption in safety-critical settings.

A principled framework for addressing this question lies in *influence functions* (IFs), a tool from robust statistics that explains how a model's predictions depend on its training data (Hampel, 1974).

This approach treats the model as an empirical entity shaped by its dataset, enabling one to trace a final prediction back to influential training points (Koh & Liang, 2017). Recent work has successfully scaled this approach to modern LLMs, demonstrating its potential to reveal generalization patterns by attributing influence down to the token level (Grosse et al., 2023). However, a key limitation persists: the IF framework assumes independence among the components of the objective (e.g., tokens in an autoregressive prediction task in prior work). This assumption is necessary for influence scores to be meaningfully interpretable, as it ensures that the relative difference in influence between components is well-defined. In practice, the tokens within LLMs are highly correlated. Thus, prior implementations, while powerful, produce influence estimates that are theoretically unsound and difficult to interpret (Basu & Echenique, 2020; Tsimpoukelli et al., 2021).

We introduce a robust framework that infers *token-level* influence on test predictions via latent mediation, enabling more reliable influence estimation. Building on recent monosemanticity research (Bricken et al., 2023; Templeton et al., 2024a) and disentangled representation learning (Wang et al., 2024), our method leverages that neural networks decompose into semantically meaningful, independent components. Our method generalizes to *general prediction tasks* by propagating influence through disentangled latent spaces where features exhibit statistical independence, critical for reliable influence estimation. Our contributions are fourfold:

1. **Unified sample- and feature-level influence**: We extend influence analysis beyond the isolated-token paradigm of prior work to model the *joint influence of tokens* within training sample-label pairs. By propagating influence from latent features to input tokens through their joint activation patterns, we attribute predictions to specific token combinations in the training data while leveraging monosemantic structure. Unlike methods treating neurons as atomic units, we recognize meaningful computation occurs at interpretable feature level spanning multiple neurons.
2. **Stable, independent feature extraction via enhanced sparse autoencoders (SAEs)**: We integrate SAEs (Gao et al., 2024; Cunningham et al., 2023; Marks et al., 2024; Cong et al., 2023) augmented with a *dynamic k-selection process* that adaptively controls sparsity during fine-tuning. Drawing from similar ideas as in disentangled representation principles (Wang et al., 2024; Chen et al., 2024), our method produces approximately independent latent features, improving the stability and interpretability of influence scores.
3. **Scalable non-decomposable influence estimation via Jacobian-vector products (JVPs)**: The shift from token-level to latent-level influence computation introduces a key challenge: unlike in autoregressive modeling, influence over latent features is holistically interdependent and does not admit additive decomposition. We develop a JVP-based method that propagates influence accurately through structured nonlinear interactions, enabling efficient and stable computation.
4. **Practical validation on medical data**: We demonstrate the utility of our framework through case studies on medical datasets, generating auditable links from model predictions back to training evidence and latent features. Connecting influence estimation to semantically meaningful features provides more actionable medical AI insights than traditional neuron-level attribution.

By unifying data-level and feature-level attribution, our approach offers a principled pathway toward transparent, trustworthy, and deployable LLMs for high-stakes domains, with additional potential for large-scale training data auditing and diagnostics, which we further discuss in Section 5. Section 2 introduces our notation and preliminaries on IF and JVP. We then describe our method in Section 3 and evaluate its performance in Section 4. Additional related works is included in Appendix A.

2 Preliminaries

Given a training dataset $\mathcal{D} = \{z_i = (x_i, y_i)\}_{i=1}^n$ i.i.d. drawn from an unknown distribution, with input $x_i \in \mathcal{X}$ and label $y_i \in \mathcal{Y}$. A model $h_\theta : \mathcal{X} \to \mathcal{Y}$ with parameters $\theta \in \mathbb{R}^p$ is trained by minimizing the empirical risk $\hat{\theta} = \arg\min_\theta \frac{1}{n} \sum_{i=1}^n \ell(h_\theta(x_i), y_i)$, where $\ell(\cdot, \cdot)$ is the loss function.

Influence Functions (IFs) In statistical estimation, the IF quantifies the sensitivity of an estimator to infinitesimal perturbations in the data, under the assumption that the data are independent. This concept extends directly to machine learning, where the high-dimensional "parameter" is the set of weights $\hat{\theta}$ of a trained neural network—a complex function of the data shaped by the architecture, loss, and optimizer. Once training is complete and the model parameters $\hat{\theta}$ are fixed, we can analyze their local sensitivity to individual training samples. This is first formalized by the *response function*,

$\hat{\theta}_{\epsilon,z_{\text{train}}}$, which describes what the optimal parameters *would be* if we were to infinitesimally upweight the loss (by ϵ) on a specific point $z_{\text{train}} = (x_{\text{train}}, y_{\text{train}})$ in the empirical risk. This perturbed objective is defined as:

$$\hat{\theta}_{\epsilon,z_{\text{train}}} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(h_\theta(x_i), y_i) + \epsilon \ell(h_\theta(x_{\text{train}}), y_{\text{train}}), \tag{1}$$

where the solution at $\epsilon = 0$ corresponds exactly to the original pre-trained parameters: $\hat{\theta}_{0,z_{\text{train}}} = \hat{\theta}$. The IF measures the sensitivity of these pre-trained parameters by computing the first-order Taylor approximation (i.e., the derivative) of the response function with respect to ϵ, at $\hat{\theta}$. Under standard regularity conditions, this can be computed using the Implicit Function Theorem (Krantz & Parks, 2002). Let $H_{\hat{\theta}} = \frac{1}{n}\sum_{i=1}^{n} \nabla^2_{\theta} \ell(h_{\hat{\theta}}(x_i), y_i)$ be the Hessian of the empirical risk evaluated at $\hat{\theta}$, then

$$\text{IF}_{\hat{\theta}}(z_{\text{train}}) = \left.\frac{d\hat{\theta}_{\epsilon,z_{\text{train}}}}{d\epsilon}\right|_{\epsilon=0} = -H_{\hat{\theta}}^{-1} \nabla_\theta \ell(h_{\hat{\theta}}(x_{\text{train}}), y_{\text{train}}). \tag{2}$$

Influential Training Samples on Test Prediction Since $\text{IF}_{\hat{\theta}}(z_{\text{train}})$ is a high-dimensional vector, it is often difficult to interpret directly. To obtain a more concrete measure, we convert this parameter-space influence into a scalar quantity by measuring its effect on a specific model output. This is done by projecting the influence vector onto the gradient of a chosen function, such as the loss or the logits for a test example $z_{\text{test}} = (x_{\text{test}}, y_{\text{test}})$. Applying the Chain Rule, we can compute the scalar *influence* of upweighting z_{train} on the loss at z_{test} as follows:

$$\mathcal{I}(z_{\text{train}}, z_{\text{test}}) = -\nabla_\theta \ell(h_{\hat{\theta}}(x_{\text{test}}), y_{\text{test}})^\top H_{\hat{\theta}}^{-1} \nabla_\theta \ell(h_{\hat{\theta}}(x_{\text{train}}), y_{\text{train}}). \tag{3}$$

This provides an interpretable measure to trace predictions back to influential training samples.

Influential Tokens on Test Prediction in Autoregressive Tasks In *autoregressive* tasks, the loss function decomposes additively across tokens, which enables the direct computation of token-level influence. This additive structure permits the gradient and Hessian in the influence function to be similarly decomposed, allowing the influence of individual training tokens to be derived explicitly. Let $\{x_1, \cdots, x_T\}$ to denote the T tokens in x_{train}. Then, Eq. (3) can be rewritten as

$$\mathcal{I}(z_{\text{train}}, z_{\text{test}}) = -\nabla_\theta \ell(h_{\hat{\theta}}(x_{\text{test}}), y_{\text{test}})^\top H_{\hat{\theta}}^{-1} \nabla_\theta \frac{1}{T} \sum_{t=1}^{T} \ell(h_{\hat{\theta}}(x_t), y_t). \tag{4}$$

Thus, the per-token influence score is defined as (Grosse et al., 2023):

$$\mathcal{I}_t(z_{\text{train}}, z_{\text{test}}) = -\nabla_\theta \ell(h_{\hat{\theta}}(x_{\text{test}}), y_{\text{test}})^\top H_{\hat{\theta}}^{-1} \frac{1}{T} \nabla_\theta \ell(h_{\hat{\theta}}(x_t), y_t). \tag{5}$$

Remark 2.1 (Problems with Existing Per-Token Influence). However, the decomposition in Eq. (5) is restricted to an autoregressive task and implicitly assumes that the tokens in each training sample are independent. This is violated in text, as tokens are highly correlated. Consequently, the influence score for a token captures not only its own effect but also the confounded effects of correlated tokens in its context. This entanglement breaks the core interpretation of the score as measuring the isolated effect of a single token, rendering the estimates unreliable. To address this, we propose augmenting the LLM with modified SAEs (Section 3), which enable influence estimation in a structured latent space where these dependencies can be better controlled.

Jacobian-Vector Products This is a key technical tool that we use. Given a function $F : \mathbb{R}^n \to \mathbb{R}^m$ and a direction $v \in \mathbb{R}^n$, the JVP at $x \in \mathbb{R}^n$ is the *directional derivative* of F at x along v:

$$\text{JVP}(F, x, v) = \left.\frac{d}{d\varepsilon} F(x + \varepsilon v)\right|_{\varepsilon=0} = J_F(x)\, v, \tag{6}$$

where $J_F(x)$ is the Jacobian of F at x. Intuitively, it answers the question: *"If I nudge the input by an infinitesimal step εv, how does the output change to first order?"*

Modern automatic differentiation libraries (e.g., PyTorch, JAX, TensorFlow) can compute JVPs directly without materializing the full Jacobian. Instead, they propagate the perturbation v forward through each primitive operation (forward-mode AD), making JVPs scalable to high-dimensional functions such as deep neural networks.

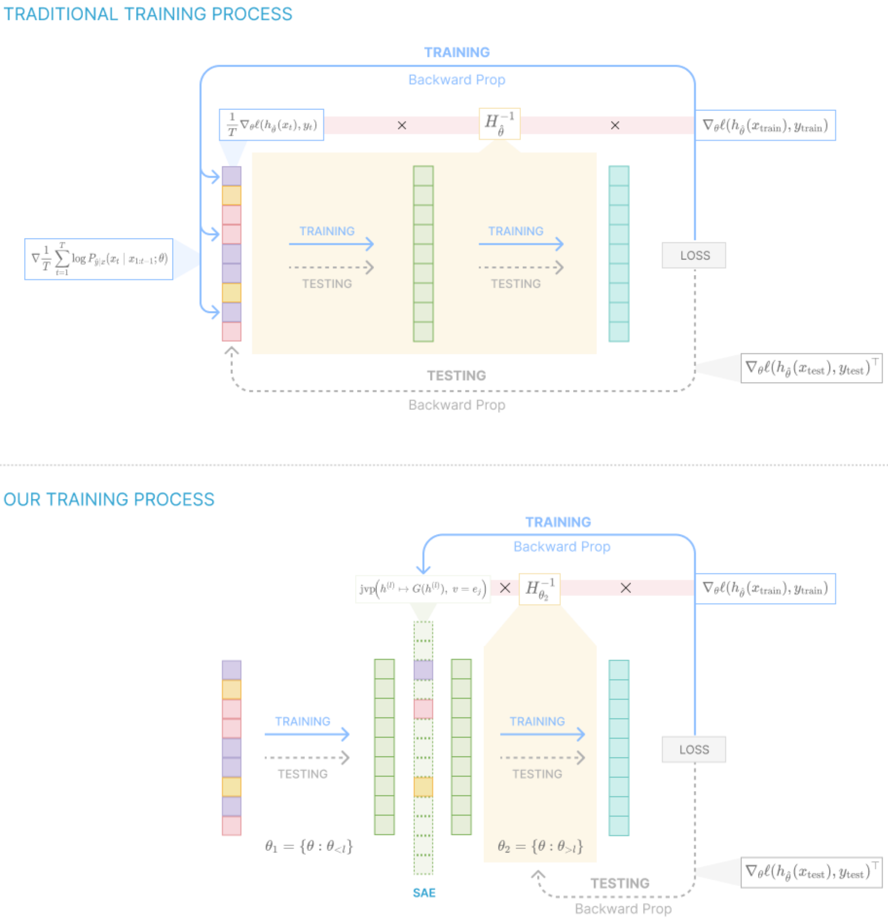

TRADITIONAL TRAINING PROCESS
TRAINING
Backward Prop
$\frac{1}{T}\nabla_\theta \ell(h_{\hat{\theta}}(x_t), y_t)$
×
$H_{\hat{\theta}}^{-1}$
×
$\nabla_\theta \ell(h_{\hat{\theta}}(x_{\text{train}}), y_{\text{train}})$
$\nabla \frac{1}{T}\sum_{t=1}^{T} \log P_{\hat{y}|x}(x_t \mid x_{1:t-1}; \theta)$
TRAINING
TESTING
TRAINING
TESTING
LOSS
$\nabla_\theta \ell(h_{\hat{\theta}}(x_{\text{test}}), y_{\text{test}})^\top$
TESTING
Backward Prop
OUR TRAINING PROCESS
TRAINING
Backward Prop
$\text{jvp}\big(h^{(l)} \mapsto G(h^{(l)}),\ v = e_j\big)$
×
$H_{\theta_2}^{-1}$
×
$\nabla_\theta \ell(h_{\hat{\theta}}(x_{\text{train}}), y_{\text{train}})$
TRAINING
TESTING
TRAINING
TESTING
LOSS
$\theta_1 = \{\theta : \theta_{<l}\}$
$\theta_2 = \{\theta : \theta_{>l}\}$
$\nabla_\theta \ell(h_{\hat{\theta}}(x_{\text{test}}), y_{\text{test}})^\top$
SAE
TESTING
Backward Prop

Figure 1: **Framework overview.** Traditional influence functions operate in the input space, assuming token independence and decomposable losses. Our method introduces a sparse autoencoder at an intermediate layer, splitting the model into upstream and downstream parts. Influence is then computed at the representation level using JVPs, enabling stable per-feature attributions and linking test predictions to interpretable sparse features.

3 Methodology

We now detail our framework that infers *token-level* influence on test predictions via a latent mediation approach, enabling more reliable influence estimation for general prediction tasks. This section presents the core components of our approach: 1) augmenting LLMs with SAEs to obtain more interpretable latent representations (Section 3.1), 2) computing influence scores over these latent features rather than directly on input tokens (Section 3.2), and 3) efficiently implementing this computation via Jacobian-vector products (Section 3.3) while maintaining the ability to propagate attributions back to the input space. Figure 1 provides an overview of the complete framework.

3.1 Augmenting LLM with Sparse Autoencoders for Independent Features

We follow Bricken et al. (2023) and Gao et al. (2024) to define a sparse autoencoder that maps input $x^l \in \mathbb{R}^d$ at layer l into a sparse latent code $r \in \mathbb{R}^h$ through

$$r = \sigma(W_{\text{enc}}(x^l - b_{\text{pre}}) + b_{\text{enc}}), \tag{7}$$

$$\tilde{x}^l = W_{\text{dec}} r + b_{\text{pre}}, \tag{8}$$

where $W_{\text{enc}} \in \mathbb{R}^{h\times d}$, $b_{\text{enc}} \in \mathbb{R}^h$, $W_{\text{dec}} \in \mathbb{R}^{d\times h}$, and $b_{\text{pre}} \in \mathbb{R}^d$. The nonlinearity $\sigma(\cdot)$ is ReLU in classical settings (Bricken et al., 2023) and TopK in modern settings (Gao et al., 2024).

However, we observe that these activation functions still lead to training instability and persistent dead latents across various sparsity regimes. To mitigate this, we introduce a **Dynamic TopK** activation function, which gradually increases sparsity constraints during training rather than imposing them abruptly at initialization. This approach gradually squeezes information into k-sparse autoencoder, promoting more stable optimization and consistently meaningful latent feature learning.

To ensure that learned sparse latent features align with domain-specific knowledge in high-stakes applications (e.g., medical data analysis), we *jointly* train the sparse autoencoder and fine-tune the model on the domain-specific dataset. Thus, our full objective function combines the reconstruction loss, sparsity penalty, and task-specific loss into a unified optimization goal:

$$\mathcal{L}_{\text{SAE}}(r) = \|x^l - \tilde{x}^l\|^2 + \lambda_1 \|r\|_1 + \lambda_2 \frac{1}{n} \sum_{i=1}^{n} \ell(h_\theta(x_i), y_i). \tag{9}$$

Let r_j be the activation of feature j, forming the basis for our feature-level influence analysis.

Remark 3.1 (SAE for Improved Influence Estimation). Classical influence functions assume independent samples, an assumption broken by token-level representations, where strong sequential correlations invalidate estimates. SAEs, by contrast, induce a latent representation comprised of approximately independent features,[1] each corresponding to a semantically meaningful concept. While these features are not strictly independent, their distributions are regularized toward comparable sparsity patterns—significantly closer to the independent structure assumptions underlying influence estimation. This alignment makes SAE-based latents far more suitable for influence score interpretation than raw token-level attribution, where strong sequential dependencies violate core requirements of the influence framework. Although influence functions technically require features to be identically distributed for comparable scaling across components, this represents a second-order concern. We expect SAEs to produce latent representations with more comparable distribution scales than raw tokens, thereby providing more reliable influence estimates.

3.2 Influence Functions on Latent Representation

Classical influence functions applied directly to correlated text tokens (Eq. (5)) are problematic due to strong sequential dependencies. To address this, we compute influence on *latent features* rather than raw tokens. As illustarated in Figure 1, we split the model parameters into two parts: $\theta = (\theta_1, \theta_2)$, where θ_1 maps a raw text input sequence x into an intermediate representation $r = h_{\theta_1}(x)$, and θ_2 maps r to the final prediction, $h_\theta(x) = h_{\theta_2}(h_{\theta_1}(x))$. Thus, the inputs to the influence function are the intermediate representations r, rather than input tokens.

Let $r_{\text{train}} = h_{\theta_1}(x_{\text{train}})$ and $r_{\text{test}} = h_{\theta_1}(x_{\text{test}})$ denote the latent representation of x_{train} and x_{test}, respectively. Define the corresponding latent-space data points as $z^r_{\text{train}} = (r_{\text{train}}, y_{\text{train}})$ and $z^r_{\text{test}} = (r_{\text{test}}, y_{\text{test}})$. The *representation-level influence function* is defined as:

$$\mathcal{I}(z^r_{\text{train}}, z^r_{\text{test}}) = -\nabla_{\theta_2} \ell(h_{\theta_2}(r_{\text{test}}), y_{\text{test}})^\top H^{-1}_{\theta_2} \nabla_{\theta_2} \ell(h_{\theta_2}(r_{\text{train}}), y_{\text{train}}). \tag{10}$$

Since θ_1 can be viewed as a deterministic projection, Eq. (10) equivalently quantifies the influence of the intermediate representation on the test point, denoted, $\mathcal{I}^r(z^r_{\text{train}}, z_{\text{test}})$, and satisfies:

$$\mathcal{I}^r(z^r_{\text{train}}, z_{\text{test}}) = \mathcal{I}(z^r_{\text{train}}, (h_{\theta_1}(x_{\text{test}}), y_{\text{test}}) = \mathcal{I}(z^r_{\text{train}}, z^r_{\text{test}}). \tag{11}$$

A key distinction emerges here: both z_{test} and z^r_{test} refer to the full test sequence, and influence is measured via its total loss. This is in contrast with Eq. (5) (Grosse et al., 2023). Now, by mapping these influential features back to the specific words that *activate* them, our explanations more faithfully capture the model's internal reasoning—moving beyond isolated token attributions toward coherent, feature-driven interpretability.

3.3 Jacobian-vector products For Non-Decomposable Losses

As noted in Remark 2.1, the additive decomposition in Eq. (5) *fails* for general prediction tasks, where gradients are defined only at the sequence level. We propose a JVP-based method that enables attribution of influence in non-decomposable settings. We demonstrate its use by quantifying how intermediate representations (and their associated training labels) influence test predictions.

[1] To promote orthogonality, we also experimented with adding an orthogonality constraint to the objective in Eq. (9). However, since it led to only negligible performance gains, we ultimately excluded this term.

Neuron-Level Influence via Perturbation Analysis Consider an intermediate activation $h^{(l)} \in \mathbb{R}^{d_l}$ at layer l, which may correspond to the output of an attention head, MLP block, or residual stream in a transformer (Elhage et al., 2021). Mechanistic interpretability studies have shown that such representations often encode semantically meaningful features causally linked to final predictions (Wang et al., 2023; Meng et al., 2022). To attribute influence to individual neurons within the latent representation r_{train} (corresponding to $h^{(l)}$), we analyze the effect of infinitesimal perturbations. Let $\theta_1 = \{\theta : \theta_{<l}\}$ and $\theta_2 = \{\theta : \theta_{>l}\}$ be the collection of parameters before and after layer l, respectively. Recall that $r_{\text{train}} = h_{\theta_1}(x_{\text{train}})$ is produced by parameters θ_1.

Eq. (11) indicates that the influence score requires two gradient terms. The first term, $g_{\text{test}} = \nabla_{\theta_2}\ell(h_{\theta_1}(x_{\text{test}}), y_{\text{test}})$, can be computed via standard backpropagation since it involves the complete forward pass. Let $r^{(j)}_{\text{train}}$ be the j-th entry contained in r_{train}. The challenge lies in the second term: computing $\nabla_{\theta_2}\ell(h_{\theta_2}(r_{\text{train}}), y_{\text{train}})$ at the *individual neuron level* (i.e., evaluated at $r^{(j)}_{\text{train}}$). We address this by introducing an infinitesimal perturbation ε to the j-th neuron of the latent representation and examining how this affects the parameter gradient:

Definition 3.2 (Neuron-Level Influence)**.** The influence score evaluated at $r^{(j)}_{\text{train}}$ is defined as:

$$\mathcal{I}^r_j(z^r_{\text{train}}, z_{\text{test}}) = -g_{\text{test}}^\top H^{-1}_{\theta_2} \frac{d}{d\varepsilon}\left(\nabla_{\theta_2}\ell(h_{\theta_2}(r_{\text{train}} + \varepsilon e_j), y_{\text{train}})\right)\Big|_{\varepsilon=0}. \tag{12}$$

This formulation quantifies how an infinitesimal perturbation to a neuron propagates through the network to influence predictions, with the sign convention ensuring that positive influence corresponds to improved train–test gradient alignment. From a mechanistic interpretability standpoint, it provides a causal, influence-based measure of each neuron's contribution, sharply contrasting with correlational metrics like activation magnitude (Koh & Liang, 2017; Geiger et al., 2021).

Jacobian-Vector Product Formulation The derivative term in Eq. (12) can be computed efficiently using JVPs, available in modern automatic differentiation frameworks (Baydin et al., 2018). This enables scalable influence estimation for large architectures, including transformers, without materializing full Jacobians. Let $G(r) = \nabla_{\theta_2}\ell(h_{\theta_2}(r), y_{\text{train}})$ denote the gradient of the loss with respect to downstream parameters θ_2, viewed as a function of an intermediate representation r. By the definition of JVP (Eq. (6)),

$$\frac{d}{d\varepsilon}G(r_{\text{train}} + \varepsilon e_j)\Big|_{\varepsilon=0} = J_G(r_{\text{train}})\, e_j, \tag{13}$$

which extracts the j-th column of the Jacobian $J_G(r_{\text{train}})$.

Thus, the JVP $J_G(r_{\text{train}})e_j$ quantifies the effect of perturbing neuron j on the downstream gradient. This provides a principled measure of sensitivity at the representation level, efficiently computable in forward mode, and naturally extends gradient-based interpretability (Simonyan et al., 2014; Sundararajan et al., 2017b) while connecting to monosemantic features discovered by sparse autoencoders (Bricken et al., 2023).

4 Experiments

We evaluate our framework on domain-specific classification and QA tasks (e.g., MedQA), focusing on three axes: (i) visualizing how a given test sample is explained by training examples and sparse features; (ii) interpreting individual neurons via joint inspection of activations and influence; (iii) quantifying independence of sparse features versus tokens.

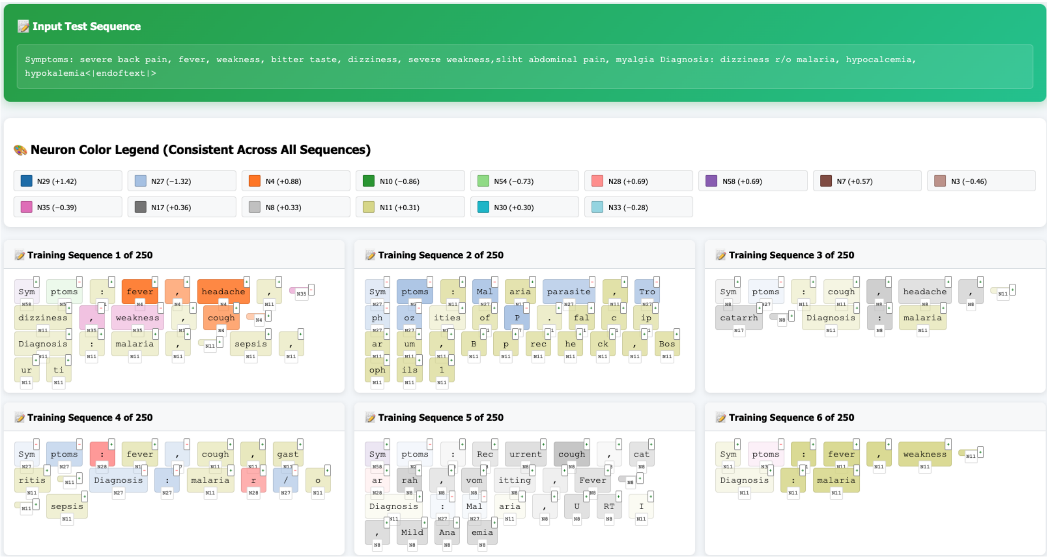
Activation–Influence Visualization We conduct experiments with `GPT-2`. At a chosen layer l, we instrument the transformer with an SAE trained on hidden states $h^{(l)}$, yielding sparse codes $r^{(l)} \in \mathbb{R}^h$. Representation-level influence is then computed as described in Section 3.

For each test sequence z_{test}, we construct an influence matrix $M_{\text{test}} \in \mathbb{R}^{|\mathcal{D}|\times h}$ capturing the influence scores between all training examples in the dataset $\mathcal{D} = \{z_i = (x_i, y_i)\}_{i=1}^n$ and each sparse feature. In parallel, we record the activation map $A_{\text{train}} \in \mathbb{R}^{L\times|\mathcal{D}|\times h}$, where L is the maximum sequence length. A_{train} is collected by running a forward pass over all training sequences and extracting layer-l activations. To obtain word-level attributions, we combine activations with influence scores. Specifically, for token j in training sequence i and neuron k, we define:

Definition 4.1 (Neuron-level Token Intensity)**.** The intensity of the j-th token in training sequence i with respect to neuron k: $\text{Intensity}(i, j, k) = M_{\text{test}}(i, k) \times A_{\text{train}}(i, j, k)$.

This construction links test predictions to both the influence of sparse features and the activations of specific training tokens. We report both the *sign* (positive or negative contribution) and the *magnitude* (strength of attribution) of these intensity scores. We made a website interaction dashboard that contains the results and we take very few samples out for analysis and walk through due to the large size of full analysis.

4.1 Neuron-Attributed Influence Visualization on AutoRegressive task



📝 Input Test Sequence

```
Symptoms: severe back pain, fever, weakness, bitter taste, dizziness, severe weakness,sliht abdominal pain, myalgia Diagnosis: dizziness r/o malaria, hypocalcemia, hypokalemia<|endoftext|>
```

Neuron Color Legend (Consistent Across All Sequences)

N29 (+1.42) | N27 (−1.32) | N4 (+0.88) | N10 (−0.86) | N54 (−0.73) | N28 (+0.69) | N58 (+0.69) | N7 (+0.57) | N3 (−0.46)
N35 (−0.39) | N17 (+0.36) | N8 (+0.33) | N11 (+0.31) | N30 (+0.30) | N33 (−0.28)

Training Sequence 1 of 250
Sym ptoms : fever , headache , dizziness , weakness , cough Diagnosis : malaria : sepsis , ur ti
Training Sequence 2 of 250
Sym ptoms : Mal aria parasite , Tro ph oz ities of P . fal c ip ar um , B p rec he ck , Bos oph ils 1
Training Sequence 3 of 250
Sym ptoms : cough , headache , catarrh Diagnosis : malaria
Training Sequence 4 of 250
Sym ptoms : fever , cough , gast ritis Diagnosis : malaria r / o sepsis
Training Sequence 5 of 250
Sym ptoms : Rec urrent cough , cat ar rah vom itting , Fever Diagnosis : Mal aria , U RT I , Mild Ana emia
Training Sequence 6 of 250
Sym ptoms : fever , weakness Diagnosis : malaria

Figure 2: The visualization of an activation influence score of a given test sample, the word-level intensity score of each training sequence, attributed by neurons. In the top right corner, the +/- mark shows whether the influence is positive or negative. The intensity of the color shows the intensity of the word. Under each word, mark the corresponding activated neuron. We deployed GPT-2on a private Symptoms-Diagnosis dataset for small-scale verification.

We begin with a qualitative case study to demonstrate how influence can be decomposed into word and neuron-level contributions in an autoregressive task: generating the text following the keyword "Diagnosis." Figure 2 illustrates one test case from a private *Symptoms–Diagnosis* dataset, analyzed with GPT-2 using our representation-level influence framework for proof of concept.

Setup Each panel in Figure 2 corresponds to a single training sequence. The sign in the top-right corner indicates whether the sequence exerts a *positive* (helpful) or *negative* (harmful) influence on the test loss. Within a sequence, tokens are shaded by their signed intensity, defined in Definition 4.1. Neurons driving these activations are annotated below each token, with consistent IDs across sequences so that feature reuse can be tracked. This decomposition is enabled by JVPs, which allow us to attribute the downstream training gradient to individual latent features.

Case study analysis The input test sequence describes a patient presenting with *severe back pain, fever, weakness, bitter taste, dizziness, abdominal pain, and myalgia*, with the diagnostic hypothesis noted as *"dizziness r/o malaria, hypocalcemia, hypokalemia"*. Our goal is to identify which training examples and sparse neurons provide the strongest support or contradiction for this test case.

Training Sequence 1 emphasizes **fever** and **headache**, highlighted strongly through neurons N4 and N34. Although "headache" is not explicitly present in the test input, fever is shared, suggesting that this sequence contributes positively through partial symptom overlap.

Training Sequence 6 highlights **malaria** directly, with neurons N11 and N8 driving strong activations. Since the test diagnosis explicitly considers "r/o malaria," this alignment indicates a positive influence: training cases where malaria co-occurs with similar upstream symptoms reinforce the diagnostic hypothesis.

Training Sequences 2-4 demonstrate negative or ambiguous contributions. For example, Sequence 2 highlights *"parasite"* and *"tropics"* as negative contributors via neuron N27. These terms may activate spurious correlations inconsistent with the current diagnostic hypothesis, pulling the model away from malaria as an explanation for dizziness. Similarly, Sequence 4 highlights irrelevant terms such as *"cough"* and *"gastritis"*, mediated by negative neurons (N27, N28).

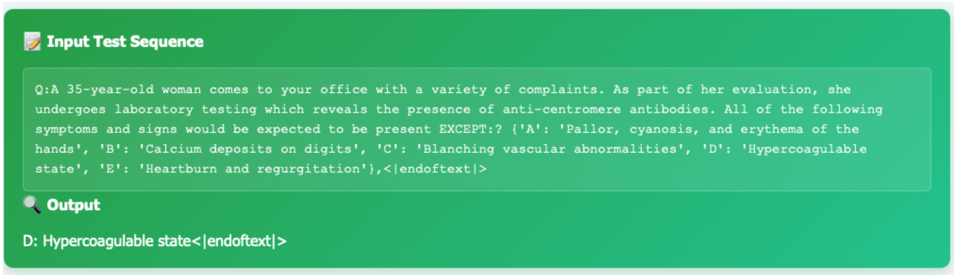
Training Sequence 5 illustrates a mixed case: *"vomiting"* and *"anemia"* are emphasized, symptoms that may co-occur with malaria but are absent from the test description. Here, neurons N28 and N8 activate moderately, yielding a weaker, more ambiguous influence signal.

Observations Across sequences, we identify several consistent patterns:

1. **Salient symptom alignment.** Tokens such as *fever*, *weakness*, and *dizziness* repeatedly emerge as hubs of positive influence, concentrated on a small set of neurons (e.g., N11, N29).
2. **Diagnostic anchoring.** Direct matches on diagnosis tokens (e.g., "malaria") yield the strongest influence, indicating that the model leverages both symptom overlap and diagnostic terms.
3. **Negative confounders.** Tokens irrelevant to the test case (e.g., *"parasite"*, *"cough"*) nonetheless elicit activations that act as negative influence, showing that the model actively downweights contradictory evidence.
4. **Neuron stability.** Certain neurons (N27/29 consistently negative/positive) recur across many sequences, pointing to stable latent axes that reliably separate supportive vs contradictory features.

This visualization shows that our framework not only recovers sequence-level influence but also decomposes it into interpretable neuron and token level contributions. The resulting patterns reveal sparse, semantically coherent features that repeatedly support/oppose predictions, thereby exposing why particular training examples are helpful or harmful for a given test case. This provides a unified framework that explains how training data shapes model behavior through interpretable latent features while uncovering semantically coherent influence patterns without architectural modifications.

4.2 Neuron-Attributed Influence Visualization on Casual Generation



Input Test Sequence

```
Q:A 35-year-old woman comes to your office with a variety of complaints. As part of her evaluation, she
undergoes laboratory testing which reveals the presence of anti-centromere antibodies. All of the following
symptoms and signs would be expected to be present EXCEPT:? {'A': 'Pallor, cyanosis, and erythema of the
hands', 'B': 'Calcium deposits on digits', 'C': 'Blanching vascular abnormalities', 'D': 'Hypercoagulable
state', 'E': 'Heartburn and regurgitation'},<|endoftext|>
```

Output

D: Hypercoagulable state<|endoftext|>

Figure 3: The test sequence we query on.

We further test our framework on the MedQA causal generation task, which requires generating the correct option from the input query.

Case study analysis The input test sequence (Figure 3) describes a *35-year-old woman* with a positive anti-centromere antibody test, asking which of several clinical features would *not* be expected. The correct answer is *"D: Hypercoagulable state"*. This provides a challenging setting where the model must connect immunological markers with systemic sclerosis features while excluding distractors. The corresponding partial result is shown in Figure 4.

Among the retrieved training sequences, we highlight three representative cases:

Training Sequence 8 emphasizes clinical features such as **dementia**, **hypertension**, and **right ankle fracture**. Tokens like *"acetaminophen"* and *"morphine"* are activated by neurons M16 and M11 with positive polarity, but these features are largely irrelevant to the current autoim-

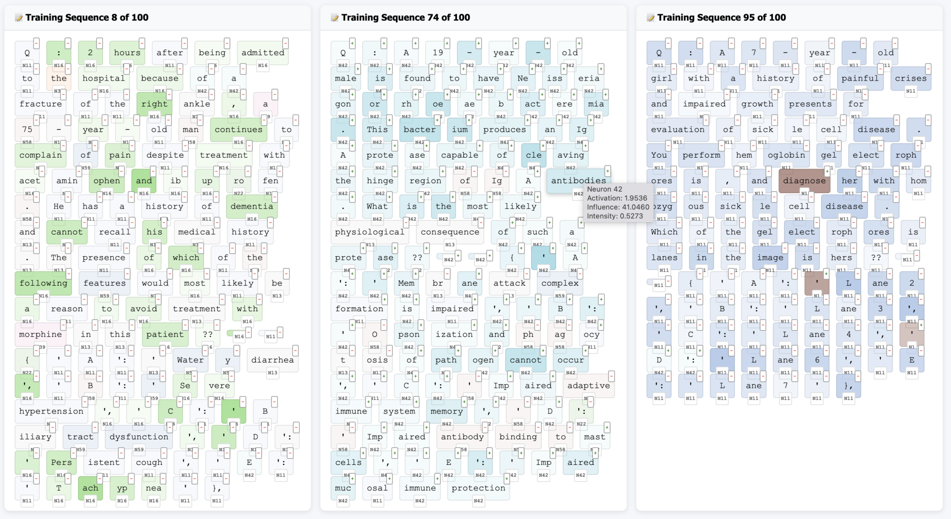

Training Sequence 8 of 100
Training Sequence 74 of 100
Training Sequence 95 of 100
Neuron 42
Activation: 1.9536
Influence: 41.0460
Intensity: 0.5273

Figure 4: Similar to Figure 2, we deployed GPT-2on a more generalized dataset MedQA.

mune/immunology context. The influence here is thus weak and noisy, showing how unrelated symptom clusters can dilute attribution.

Training Sequence 74 is strongly aligned with the test case: words such as **antibodies**, **immune**, and **pathogen** are highlighted by neuron N42, which carries a strong positive influence. This overlap reflects the immunological domain of the test case, anchoring the decision on relevant biomedical features. The visualization shows concentrated activation on immune-related terms, suggesting that the model reuses stable "immune response" neurons across contexts.

Training Sequence 95 highlights terms like **diagnose**, **sickle cell disease**, and **painful crises**, mediated primarily by neuron N11 with negative polarity. Although these are clear medical features, they represent a different disease context (hematology rather than autoimmune disease). The negative contribution indicates that the model correctly downweights evidence from sickle cell related cases when reasoning about anti-centromere antibodies.

Observations. From this visualization, we draw several insights:

1. **Domain alignment.** Positive influence concentrates on immunology-related training cases, with stable neurons (e.g., N42) capturing antibody and immune system terminology.
2. **Filtering confounders.** Consistent across datasets.
3. **Sparse concentration.** Only a handful of neurons (N42, N11, M16) carry most of the influence mass, supporting the view that sparse latent axes serve as interpretable mediators of attribution.
4. **Consistency across context.** Neurons encoding immune features appear across multiple sequences and consistently act as positive contributors, suggesting robust latent semantics.

The MedQA visualization confirms that our method can surface domain-relevant training support, while also identifying negatively influential confounders (e.g., sickle-cell disease). Influence is mediated by a small, stable set of neurons, reinforcing the interpretability and sparsity of our approach. Due to limitation of space, more results will be demonstrated in supplementary materials.

5 Discussion and Conclusion

Our framework extends influence functions beyond training–test attribution to reveal *why* examples matter, via sparse latent features, offering a dual view that exposes supportive versus harmful training evidence. While these initial results, based on a GPT-2model, demonstrate a key implication for **data quality control**—showing that harmful influence often arises from spurious examples—their scope is necessarily constrained. Implementing this framework on larger-scale LLMs like LLaMA-3.1-1Bis a compelling future direction, but remains beyond the current computational scope. Therefore, these findings should be viewed as a promising proof-of-concept, pointing toward a practical pathway for dataset refinement that warrants further validation at scale.

References

Ahmed Abdulaal, Hugo Fry, Nina Montaña-Brown, Ayodeji Ijishakin, Jack Gao, Stephanie Hyland, Daniel C Alexander, and Daniel C Castro. An x-ray is worth 15 features: Sparse autoencoders for interpretable radiology report generation. *arXiv preprint arXiv:2410.03334*, 2024.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *CoRR*, 2024. doi: 10.48550/ARXIV.2406.11717. URL https://arxiv.org/abs/2406.11717.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):e0130140, July 2015. ISSN 1932-6203. doi: 10.1371/journal.pone.0130140. URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0130140.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. *CoRR*, June 2021. doi: 10.48550/arXiv.2106.07682. URL http://arxiv.org/abs/2106.07682.

Fazl Barez, Tung-Yu Wu, Iván Arcuschin, Michael Lan, Vincent Wang, Noah Siegel, Nicolas Collignon, Clement Neo, Isabelle Lee, Alasdair Paren, Adel Bibi, Robert Trager, Damiano Fornasiere, John Yan, Yanai Elazar, and Yoshua Bengio. Chain-of-thought is not explainability, 2025.

Pathikrit Basu and Federico Echenique. On the falsifiability and learnability of decision theories. *Theoretical Economics*, 15(4):1279–1305, 2020.

Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018. URL http://jmlr.org/papers/v18/17-468.html.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7.

Blair Bilodeau, Natasha Jaques, Pang Wei Koh, and Been Kim. Impossibility theorems for feature attribution. *Proc. Natl. Acad. Sci. U.S.A.*, 121(2):e2304406120, January 2024. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2304406120. URL http://arxiv.org/abs/2212.11870.

Sid Black, Lee Sharkey, Leo Grinsztajn, Eric Winsor, Dan Braun, Jacob Merizian, Kip Parker, Carlos Ramón Guevara, Beren Millidge, Gabriel Alfour, and Connor Leahy. Interpreting neural networks through the polytope lens. *CoRR*, November 2022. URL https://arxiv.org/abs/2211.12312.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *ICLR*, 2023. URL http://arxiv.org/abs/2212.03827.

Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024.

Nick Cammarata, Gabriel Goh, Shan Carter, Ludwig Schubert, Michael Petrov, and Chris Olah. Curve detectors. *Distill*, June 2020. URL https://distill.pub/2020/circuits/curve-detectors.

Nick Cammarata, Gabriel Goh, Shan Carter, Chelsea Voss, Ludwig Schubert, and Chris Olah. Curve circuits. *Distill*, 2021. URL https://distill.pub/2020/circuits/curve-circuits/.

Giuseppe Casalicchio, Christoph Molnar, and Bernd Bischl. Visualizing the feature importance for black box models. *ECML PKDD*, 11051:655–670, 2018. doi: 10.1007/978-3-030-10925-7_40. URL http://arxiv.org/abs/1804.06620.

Lawrence Chan, Leon Lang, and Erik Jenner. Natural abstractions: Key claims, theorems, and critiques. *AI Alignment Forum*, March 2023. URL https://www.alignmentforum.org/posts/gvzW46Z3BsaZsLc25/natural-abstractions-key-claims-theorems-and-critiques-1.

David Chanin, Anthony Hunter, and Oana-Maria Camburu. Identifying linear relational concepts in large language models. *CoRR*, 2023. doi: 10.48550/ARXIV.2311.08968. URL https://arxiv.org/abs/2311.08968.

Guangyi Chen, Yifan Shen, Zhenhao Chen, Xiangchen Song, Yuewen Sun, Weiran Yao, Xiao Liu, and Kun Zhang. Learning disentangled representation for multi-modal time-series sensing signals. *Proceedings of the ACM on Web Conference 2025*, 2024.

Lin William Cong, Guanhao Feng, Jingyu He, and Junye Li. Sparse modeling under grouped heterogeneity with an application to asset pricing. Technical report, National Bureau of Economic Research, 2023.

Ian C. Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: a unified framework for model explanation. *J. Mach. Learn. Res.*, 22(1):209:9477–209:9566, January 2021. ISSN 1532-4435. URL https://arxiv.org/abs/2011.14878.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Mingyang Deng, Lucas Tao, and Joe Benton. Measuring feature sparsity in language models. *CoRR*, 2023. doi: 10.48550/ARXIV.2310.07837. URL https://arxiv.org/abs/2310.07837.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable LLM feature circuits. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=J6zHcScAo0.

N Elhage, N Nanda, C Olsson, T Henighan, N Joseph, B Mann, A Askell, Y Bai, A Chen, T Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

Nelson Elhage, Tristan Hume, Olsson Catherine, Nanda Neel, Tom Henighan, Scott Johnston, Sheer ElShowk, Nicholas Joseph, Nova DasSarma, Ben Mann, Danny Hernandez, Amanda Askell, Kamal Ndousse, Dawn Drain, Anna Chen, Yuntao Bai, Deep Ganguli, Liane Lovitt, Zac Hatfield-Dodds, Jackson Kernion, Tom Conerly, Shauna Kravec, Stanislav Fort, Saurav Kadavath, Josh Jacobson, Eli Tran-Johnson, Jared Kaplan, Jack Clark, Tom Brown, Sam McCandlish, Dario Amodei, and Christopher Olah. Softmax linear units. *Transformer Circuits Thread*, 2022a. URL https://transformer-circuits.pub/2022/solu/index.html.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *Transformer Circuits Thread*, 2022b. URL https://transformer-circuits.pub/2022/toy_model/index.html.

Nelson Elhage, Robert Lasenby, and Christopher Olah. Privileged bases in the transformer residual stream. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/privileged-basis/index.html.

Joshua Engels, Isaac Liao, Eric J. Michaud, Wes Gurnee, and Max Tegmark. Not all language model features are linear. *CoRR*, May 2024. doi: 10.48550/arXiv.2405.14860. URL http://arxiv.org/abs/2405.14860.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.

Joseph Futoma, Morgan Simons, Trishan Panch, Finale Doshi-Velez, and Leo Anthony Celi. The myth of generalisability in clinical research and machine learning in health care. *The Lancet Digital Health*, 2(9):e489–e492, 2020.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. *NeurIPS*, 34:9574–9586, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/4f5c422f4d49a5a807eda27434231040-Abstract.html.

Marzyeh Ghassemi, Luke Oakden-Rayner, and Andrew L Beam. The false hope of current approaches to explainable artificial intelligence in health care. *The lancet digital health*, 3(11): e745–e750, 2021.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, et al. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*, 2023.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *TMLR*, 2023. URL https://arxiv.org/abs/2305.01610.

Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393, 1974.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. *EMNLP*, October 2023. doi: 10.48550/arXiv.2310.15916. URL http://arxiv.org/abs/2310.15916.

Tom Henighan, Shan Carter, Tristan Hume, Nelson Elhage, Robert Lasenby, Stanislav Fort, Nicholas Schiefer, and Christopher Olah. Superposition, memorization, and double descent. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/toy-double-descent/index.html.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. Linearity of relation decoding in transformer language models. *CoRR*, August 2023. doi: 10.48550/arXiv.2308.09124. URL http://arxiv.org/abs/2308.09124.

Aapo Hyvärinen. Independent component analysis: recent advances. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110534, 2013.

M. Ivanitskiy, Alexander F. Spies, Tilman Rauker, Guillaume Corlouer, Chris Mathwin, Lucia Quirke, Can Rager, Rusheb Shah, Dan Valentine, Cecilia Diniz Behn, Katsumi Inoue, and Samy Wu Fung. Structured world representations in maze-solving transformers. *CoRR*, December 2023. URL https://arxiv.org/abs/2312.02566.

Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

janus. Simulators. *LessWrong*, September 2022. URL https://www.lesswrong.com/posts/vJFdjigzmcXMhNTsx/simulators.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

Adam Karvonen. Emergent world models and latent variable estimation in chess-playing language models. *COLM*, July 2024. doi: 10.48550/arXiv.2403.15498. URL http://arxiv.org/abs/2403.15498.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. *ICML*, July 2019. doi: 10.48550/arXiv.1905.00414. URL http://arxiv.org/abs/1905.00414.

Steven George Krantz and Harold R Parks. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2002.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. *ICLR*, 2023. URL https://arxiv.org/abs/2210.13382.

Alireza Makhzani and Brendan Frey. K-sparse autoencoders. *arXiv preprint arXiv:1312.5663*, 2013.

Giovanni Luca Marchetti, Christopher Hillar, Danica Kragic, and Sophia Sanborn. Harmonics of learning: Universal fourier features emerge in invariant networks. *CoRR*, December 2023. doi: 10.48550/arXiv.2312.08550. URL http://arxiv.org/abs/2312.08550.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*, 2024.

Callum McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. Copy suppression: Comprehensively understanding an attention head. *CoRR*, October 2023. doi: 10.48550/arXiv.2310.04625. URL http://arxiv.org/abs/2310.04625.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *NeurIPS*, 2022. doi: 10.48550/arXiv.2202.05262. URL http://arxiv.org/abs/2202.05262.

Neel Nanda. 200 cop in mi: Interpreting algorithmic problems. *Neel Nanda's Blog*, 2022. URL https://www.lesswrong.com/posts/ejtFsvyhRkMofKAFy/200-cop-in-mi-interpreting-algorithmic-problems.

Neel Nanda. Actually, othello-gpt has a linear emergent world representation. *Neel Nanda's Blog*, March 2023. URL https://neelnanda.io/mechanistic-interpretability/othello.

Michael Oberst and David Sontag. Counterfactual off-policy evaluation with gumbel-max structural causal models. In *International Conference on Machine Learning*, pp. 4881–4890. PMLR, 2019.

Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, March 2018. URL https://distill.pub/2018/building-blocks.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Laura O'Mahony, Vincent Andrearczyk, Henning Muller, and Mara Graziani. Disentangling neuron representations with concept vectors. *CVPR Workshops*, April 2023. doi: 10.48550/arXiv.2304.09707. URL http://arxiv.org/abs/2304.09707.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. *ICML MI Workshop (Oral)*, June 2024. URL https://openreview.net/forum?id=KXuYjuBzKo.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Improving dictionary learning with gated sparse autoencoders. *CoRR*, April 2024. doi: 10.48550/arXiv.2404.16014. URL http://arxiv.org/abs/2404.16014.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. *NAACL*, August 2016. doi: 10.48550/arXiv.1602.04938. URL http://arxiv.org/abs/1602.04938.

Adam Scherlis, Kshitij Sachan, Adam S. Jermyn, Joe Benton, and Buck Shlegeris. Polysemanticity and capacity in neural networks. *CoRR*, July 2023. doi: 10.48550/arXiv.2210.01892. URL http://arxiv.org/abs/2210.01892.

Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *ICCV*, 2016. URL https://arxiv.org/abs/1610.02391.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. Role play with large language models. *Nature*, 623(7987):493–498, November 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06647-8. URL https://www.nature.com/articles/s41586-023-06647-8.

Lee Sharkey. Circumventing interpretability: How to defeat mind-readers. *CoRR*, December 2022. doi: 10.48550/ARXIV.2212.11415. URL https://arxiv.org/abs/2212.11415.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *ICML*, 2017. doi: 10.48550/arXiv.1704.02685. URL http://arxiv.org/abs/1704.02685.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, April 2014. doi: 10.48550/arXiv.1312.6034. URL http://arxiv.org/abs/1312.6034.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, June 2017. doi: 10.48550/arXiv.1706.03825. URL http://arxiv.org/abs/1706.03825.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017a.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *ICML*, June 2017b. doi: 10.48550/arXiv.1703.01365. URL http://arxiv.org/abs/1703.01365.

Viacheslav Surkov, Chris Wendler, Mikhail Terekhov, Justin Deschenaux, Robert West, and Caglar Gulcehre. Unpacking sdxl turbo: Interpreting text-to-image models with sparse autoencoders. In *Mechanistic Interpretability for Vision at CVPR 2025 (Non-proceedings Track)*, 2025.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, and Brian Chen. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024a. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024b. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. Language models linearly represent sentiment. *ICML MI Workshop*, June 2024. URL https://openreview.net/forum?id=Xsf6dOOMMc.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. *CoRR*, 2023. doi: 10.48550/ARXIV.2310.15213. URL https://arxiv.org/abs/2310.15213.

Eric J Topol. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine*, 25(1):44–56, 2019.

Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965, 2023.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *ICLR*, 2023. doi: 10.48550/arXiv.2211.00593. URL http://arxiv.org/abs/2211.00593.

Xin Wang, Hong Guo, Sumit Jha, and Ruishan Gao. Disentangled representation learning. *arXiv preprint arXiv:2211.11695*, 2024.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020. doi: 10.1162/tacl_a_00321. URL https://aclanthology.org/2020.tacl-1.25.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*, 2019.

Qingyu Yin, Chak Tou Leong, Hongbo Zhang, Minjun Zhu, Hanqi Yan, Qiang Zhang, Yulan He, Wenjie Li, Jun Wang, Yue Zhang, et al. Direct preference optimization using sparse feature-level constraints. *arXiv preprint arXiv:2411.07618*, 2024.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency. *CoRR*, October 2023. doi: 10.48550/arXiv.2310.01405. URL http://arxiv.org/abs/2310.01405.

Appendix

A Related Works

Interpretability in LLM Interpretability methods range from black-box approaches like perturbation and sensitivity analysis (Casalicchio et al., 2018; Ribeiro et al., 2016; Covert et al., 2021; Warstadt et al., 2020), to gradient-based attribution methods (Smilkov et al., 2017; Sundararajan et al., 2017a; Bach et al., 2015; Shrikumar et al., 2017; Selvaraju et al., 2016; Bilodeau et al., 2024), and concept-based representations probing (Belinkov, 2022; Kornblith et al., 2019; Bansal et al., 2021; Burns et al., 2023; Zou et al., 2023; Arditi et al., 2024). More recent work in mechanistic interpretability focuses on reverse-engineering internal model structures through circuit analysis (Olah et al., 2018; Elhage et al., 2021; 2022b), and feature discovery (Bricken et al., 2023; Sharkey, 2022; Cunningham et al., 2023; Deng et al., 2023). In addition to monosemanticity and disentanglement, this line of work has enabled analyses of motifs like induction heads or copy suppression (Olsson et al., 2022; McDougall et al., 2023; Cammarata et al., 2020; 2021), universality (Chan et al., 2023; Gurnee et al., 2023; Marchetti et al., 2023), and emergent world models (Li et al., 2023; Nanda, 2023; Ivanitskiy et al., 2023; Karvonen, 2024; Shanahan et al., 2023; janus, 2022). Unlike these approaches, which often prioritize global model understanding, our method emphasizes actionable, testable attributions tailored for high-stakes domains like healthcare, where rapid fact-checking and validation of model decisions are critical for reliability and trust.

Sparse Autoencoders and Independent Features SAEs learn disentangled, interpretable features via sparsity constraints (e.g., L1 penalty), promoting statistical independence in latent representations. This approach builds upon a long history of seeking independent data components, including classical linear methods like Principal Component Analysis (PCA) (Jolliffe & Cadima, 2016) and Independent Component Analysis (ICA) (Hyvärinen, 2013), as well as nonlinear probabilistic frameworks like Variational Autoencoders (VAEs) (Kingma & Welling, 2014). However, SAEs offer a uniquely transparent and deterministic pathway to feature learning that balances sparsity and reconstruction fidelity. They are widely used for mechanistic interpretability in LLMs (Cunningham et al., 2023; Bricken et al., 2023; Templeton et al., 2024b; Marks et al., 2024), with variants including k-sparse SAEs (Makhzani & Frey, 2013), gated and JumpReLU SAEs (Rajamanoharan et al., 2024), and TopK methods (Gao et al., 2024; Bussmann et al., 2024). Beyond language, SAEs extend to multimodal domains Surkov et al. (2025), radiology and medical imaging (Abdulaal et al., 2024), and reinforcement learning alignment (Yin et al., 2024), demonstrating versatility across tasks. Recent work shows that transcoders (which approximate dense MLP behavior via wider, sparsely-activating networks) often match or exceed SAEs in interpretability and fidelity (Dunefsky et al., 2024). Extending our framework to handle independent logits from a transcoder is promising but beyond the scope of this work.

Monosemanticity and Disentanglement The pursuit of monosemantic features, where neurons respond to single coherent concepts, represents a major focus in interpretability research. This effort addresses the phenomenon of polysemanticity, explained through the superposition hypothesis (Olah et al., 2018; Elhage et al., 2021; 2023; Scherlis et al., 2023; Henighan et al., 2023). Solutions include both architectural modifications such as k-sparse autoencoders (Makhzani & Frey, 2013), softmax linear units (Elhage et al., 2022a; Rajamanoharan et al., 2024), as well as post-hoc methods like SAEs (Bricken et al., 2023; Sharkey, 2022; Cunningham et al., 2023; Deng et al., 2023). Studies have examined the linearity of representations (Nanda, 2022; Engels et al., 2024; O'Mahony et al., 2023; Hendel et al., 2023; Todd et al., 2023; Hernandez et al., 2023; Chanin et al., 2023; Tigges et al., 2024; Arditi et al., 2024), identified counterexamples such as circular features (Engels et al., 2024) and non-linear perspectives (Black et al., 2022). Geometry-aware analyses show structured organization (Park et al., 2024), and scaling studies (Templeton et al., 2024b) suggest disentanglement improves with model size. While these works aim for complete monosemanticity, our approach uses SAEs to obtain approximately independent features specifically to enable more reliable influence estimation, prioritizing practical interpretability over full disentanglement.

B LLM Usage

LLMs were used in preparing this manuscript. Their use was limited to minor editorial polishing of wording and style. All conceptual development, methodology, and results are original and are fully described in the paper.

C Reproducibility Statement

We have taken several steps to ensure the reproducibility of our results. All methods are described in detail in Section 3, including the formulation of representation-level influence, the use of Jacobian vector products, and integration with sparse autoencoders. Experimental setups, datasets, and model checkpoints are documented in Section 4. Visualizations (Figures 2, 4) are generated through scripts will be released as part of our anonymous code submission as supplementary material and will release it publicly upon publication.