Sparse Feature Circuits

Discovering and editing INTERPRETABLE CAUSAL GRAPHS IN LANGUAGE MODELS

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, Aaron Mueller

International Conference of Learning Representations (ICLR) 2025' Oral

Presentation By Ethan Nguyen and Matthew Majeske

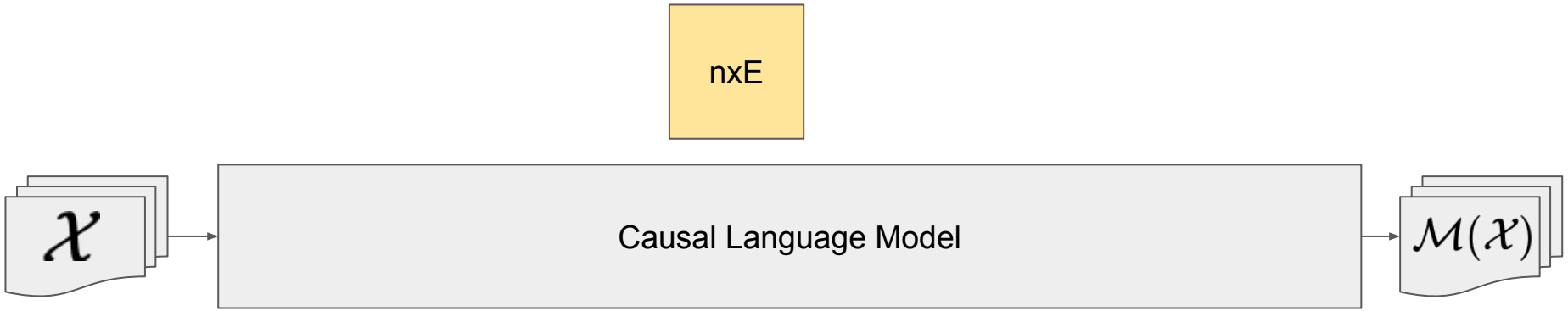
Priming Questions

- Is it possible to fundamentally change how a model 'reasons' about a concept without showing it a single new piece of training data?
- If we 'cut out' a specific thought from a model's brain, do we lose the 'intelligence' surrounding it, or can we truly isolate a single unintended signal?
- If Fine-tuning (RLHF, SFT, RLVR, RLAIF) can teach a model to act a certain way towards us, how do we know if the bias is actually gone—or if the model has just learned to hide natural propensities?

$\mathcal{M} : \mathcal{T} -> \mathbb{R}^E$

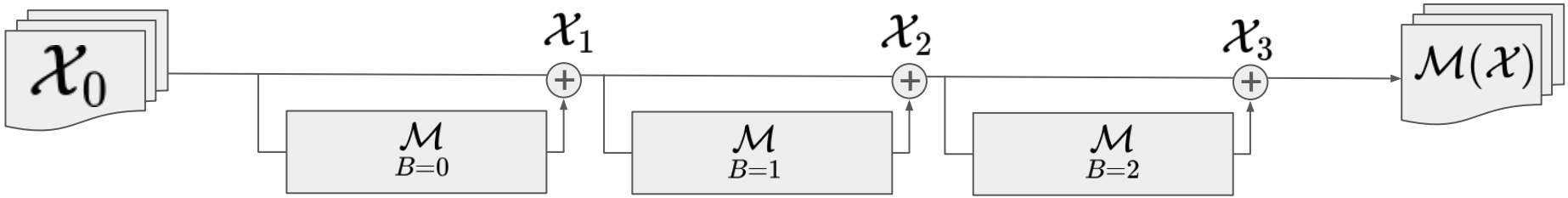
Understanding LLM Notation & Layer Indexing

We define our causal language model as $\mathcal{M} : \mathcal{T} -> \mathbb{R}^E$. Generally we consider our models input as $\mathcal{X} = [x_0, \ldots, x_n]$, where $\mathcal{X} \in \mathcal{T}$. Our models output is generally defined as $\mathcal{M}(\mathcal{X}) \in \mathbb{R}^{n \times E}$



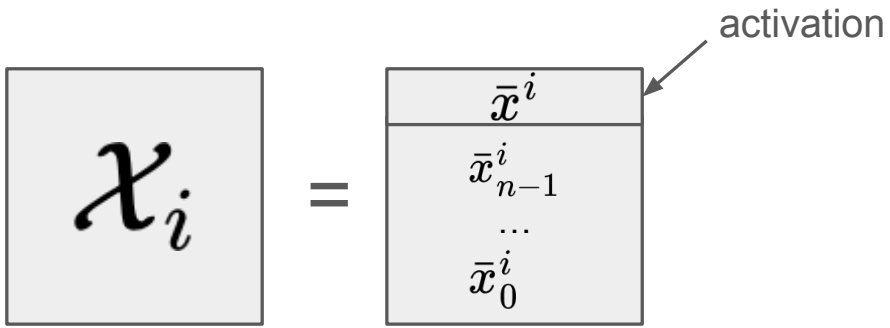
Target Layers & Intermediate Outputs

We use the the notation $\underset{B=1}{\mathcal{M}}$ where B is arbitrary block within the LM like a transformer block. We define the i-th block's output as $\mathcal{X}_{i+1} = \underset{B=i}{\mathcal{M}}(\mathcal{X}_i, \mathcal{X}) \equiv \underset{B=i}{\mathcal{M}}(\mathcal{X}_i)$ using the aforementioned recursive definition.



Difference between an activation and an intermediate output

In our works we refer to an activation $\bar{x}^i$ as the n-th vector of an intermediate output. What this means is that it captures the entirety of the previous token information. We call our activation our output $\bar{x}^i \in R^E$ which corresponds to a singular vector.



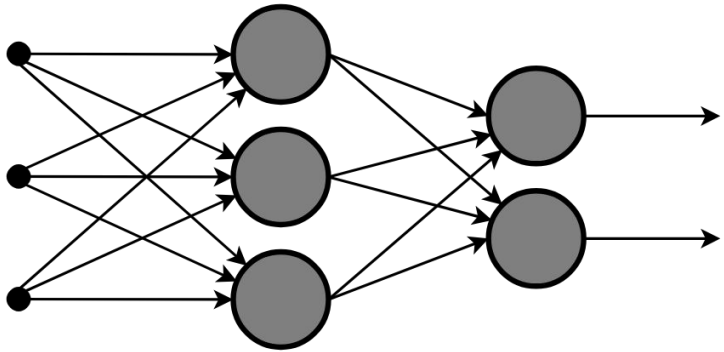
Polysemanticity & the Superposition Hypothesis

Neurons produce activations, $\bar{x}^i \in R^E$

$$\mathcal{K} < E \ll \#\text{ of concepts}$$

Kissing #

Therefore, each neuron must be polysemantic
ie, corresponding to more than one concept



What is a Feature? (Theory to Implementation)

Properties of a feature, f from $\mathcal{M}$:

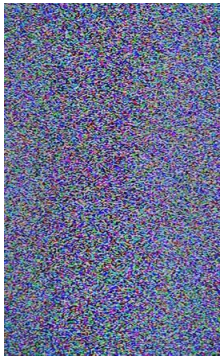
1. Correspond to useful information for computation
2. Monosemantic
3. Fires when only active

$$f(\bar{x}) = ReLU(d \cdot \bar{x} + b)$$

Human interpretable feature ie uncertainty in response

Learned weighting of the internal feature representation to composes into monosemantic human understandable feature

Model's Internal feature representation



How do we find monosemantic neuron?

What if we use AI to understand AI? We introduce the sparse autoencoder, which is an architecture meant to derive monosemantic features from a polysemantic activation $W_{dec}, W_{enc}^{\top} \in \mathbb{R}^{d_{in} \times Q} \quad d_{in} \ll Q$

$$W_{\text{enc}} = [d_1, d_2, \ldots, d_q]^T$$

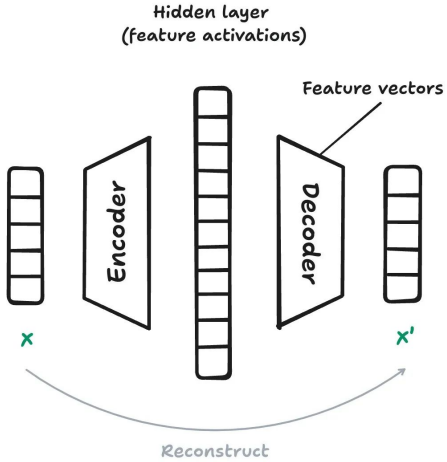


Figure Adapted from LessWrong Contributor Nick Jiang 2024

$$\mathscr{L}(\bar{x}, \lambda) = ||\bar{x} - SAE(\bar{x})||_2^2 + \lambda ||f(\bar{x})||_0$$

$$\begin{aligned} SAE(\bar{x}) &= W_{dec} ReLU(W_{enc}\bar{x} + b) \\ &= W_{dec} f(\bar{x}) \end{aligned}$$

Note: There's a lot of SAE architectures

Neuronpedia (auto interpretability)

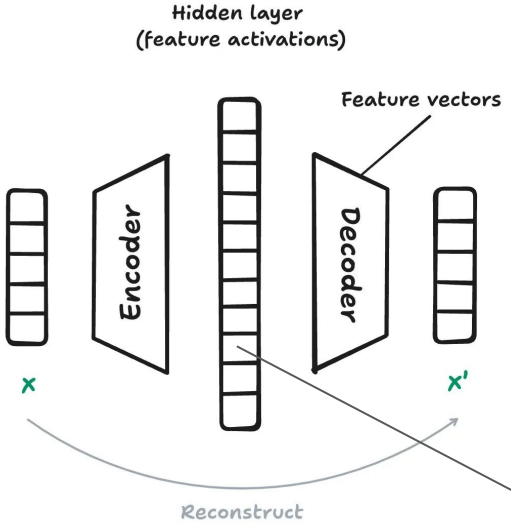


Figure Adapted from LessWrong Contributor Nick Jiang 2024

Carolina 40.82	future of eastern North Carolina and the counties along
Carolina 40.69	Kitty Hawk , North Carolina . In the two
Carolina 40.60	decision of the North Carolina Court of Appeals pursuant
Carolina 40.51	popular vote , North Carolina would be Henry Clay
Carolina 40.46	appeal from the North Carolina Court of Appeals ,
Carolina 38.02	in Charleston , South Carolina , I
tomato 36.44	cheddar cheese , fresh tomato salsa and pumpkin pie

references to North Carolina

Fig from NeuronPedia Gemma Scope / Layer 1 Residual Stream / Index 14675

Problem Statement

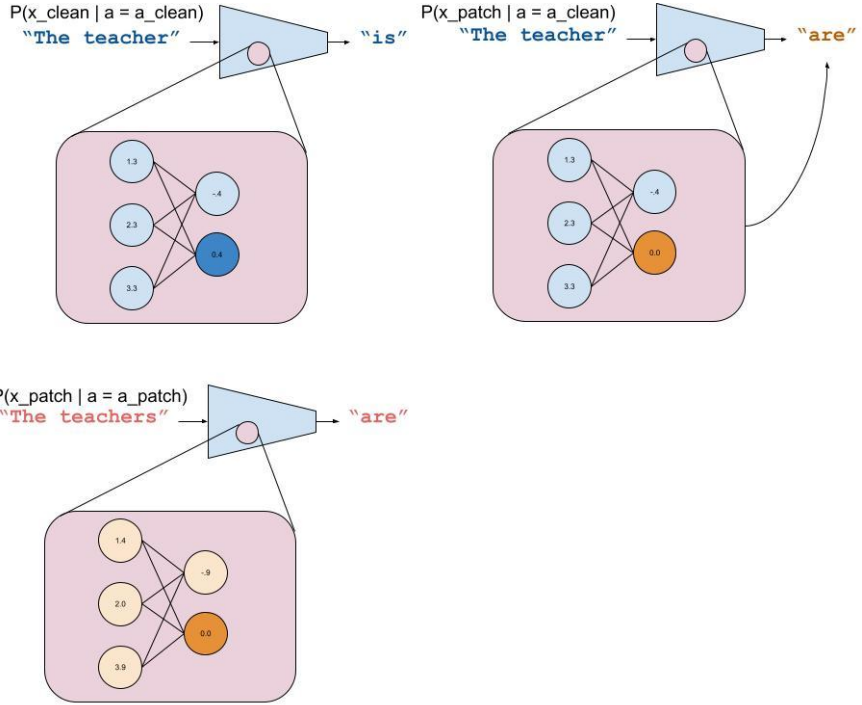
We want to control computational mechanisms within models to remove underlying biases or misconceptions about the task or the world. We want our mechanism to interpretable to humans and monosemantic.

Indirect Effect Metric

The amount of influence a neuron has on the model's output decision based on respective inputs.

m = a metric that scores how much the model prefers out_clean over out_patch (or vice versa)

$$\hat{m}(x) = \log P(\text{“are”}|x) - \log P(\text{“is”}|x)$$



Linear Approximations:

$$\hat{\text{IE}}_{\text{atp}}(m; \mathbf{a}; x_{\text{clean}}, x_{\text{patch}}) = \nabla_{\mathbf{a}} m|_{\mathbf{a}=\mathbf{a}_{\text{clean}}} (\mathbf{a}_{\text{patch}} - \mathbf{a}_{\text{clean}})$$

$$\hat{\text{IE}}_{\text{ig}}(m; \mathbf{a}; x_{\text{clean}}, x_{\text{patch}}) = \frac{1}{N} \left(\sum_{\alpha} \nabla_{\mathbf{a}} m|_{\alpha \mathbf{a}_{\text{clean}} + (1-\alpha)\mathbf{a}_{\text{patch}}} \right) (\mathbf{a}_{\text{patch}} - \mathbf{a}_{\text{clean}})$$

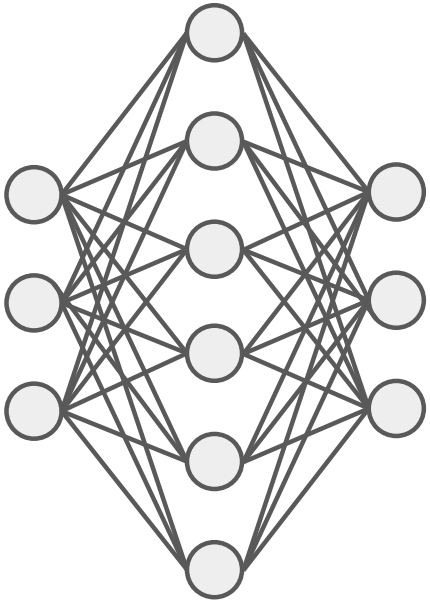
$$\text{IE}(m; \mathbf{a}; x_{\text{clean}}, x_{\text{patch}}) = m\left(x_{\text{clean}}|\text{do}(\mathbf{a} = \mathbf{a}_{\text{patch}})\right) - m(x_{\text{clean}})$$

These methods can be parallelized using taylor series expansion

Building a Scalpel

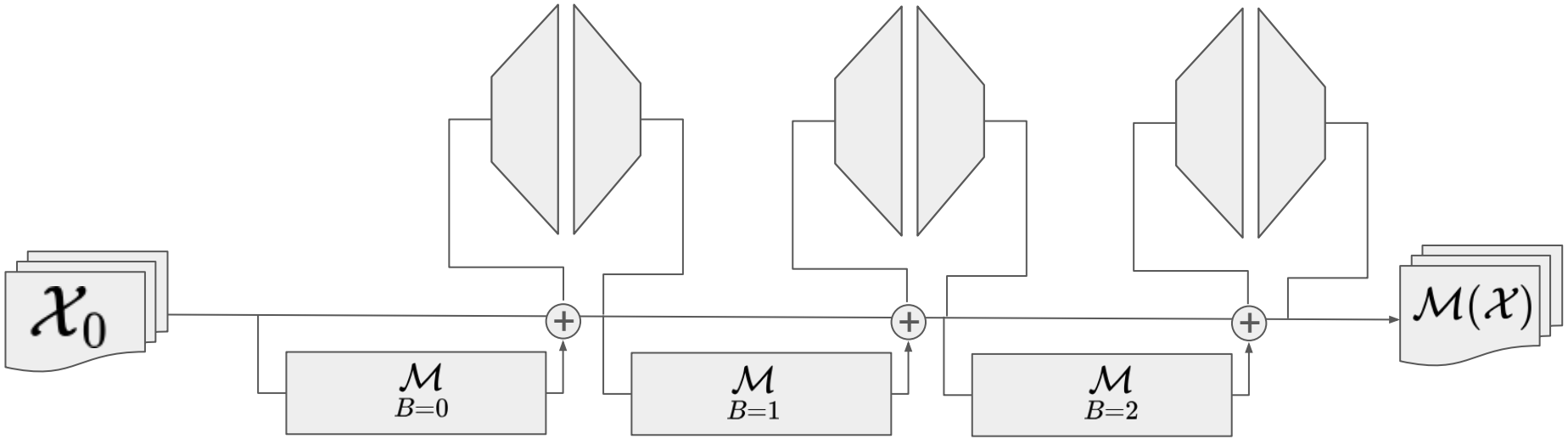
SAE intermediate features are mostly monosemantic and correspond to one idea! The original input is equivalent to the output + some error

$$\bar{x} = SAE(\bar{x}) + \epsilon$$



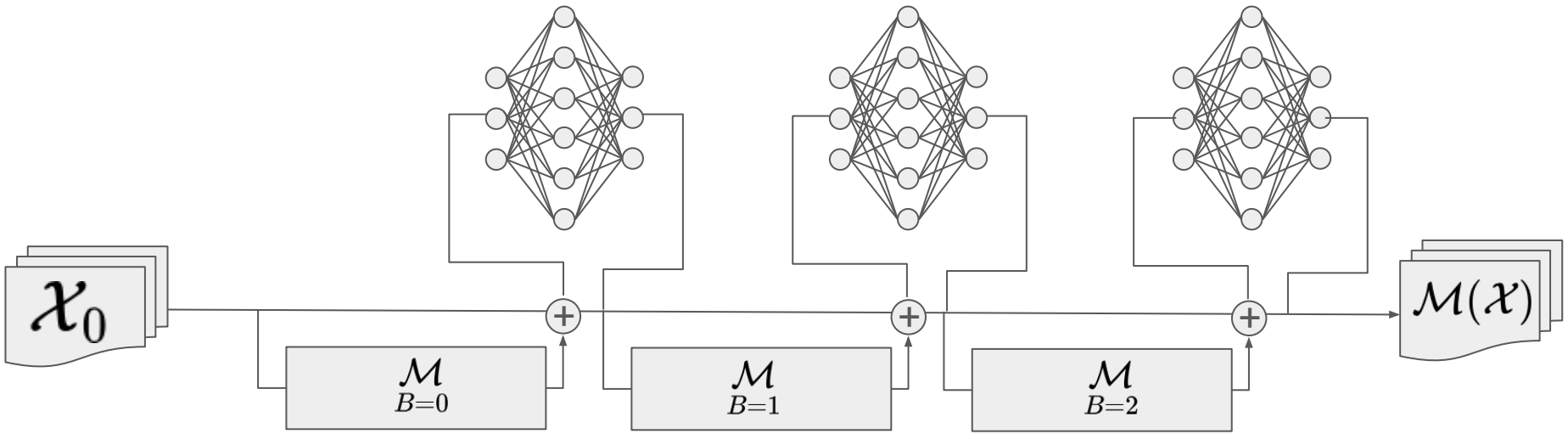
Sparse Feature Circuits

Injected SAE into the model computational graph

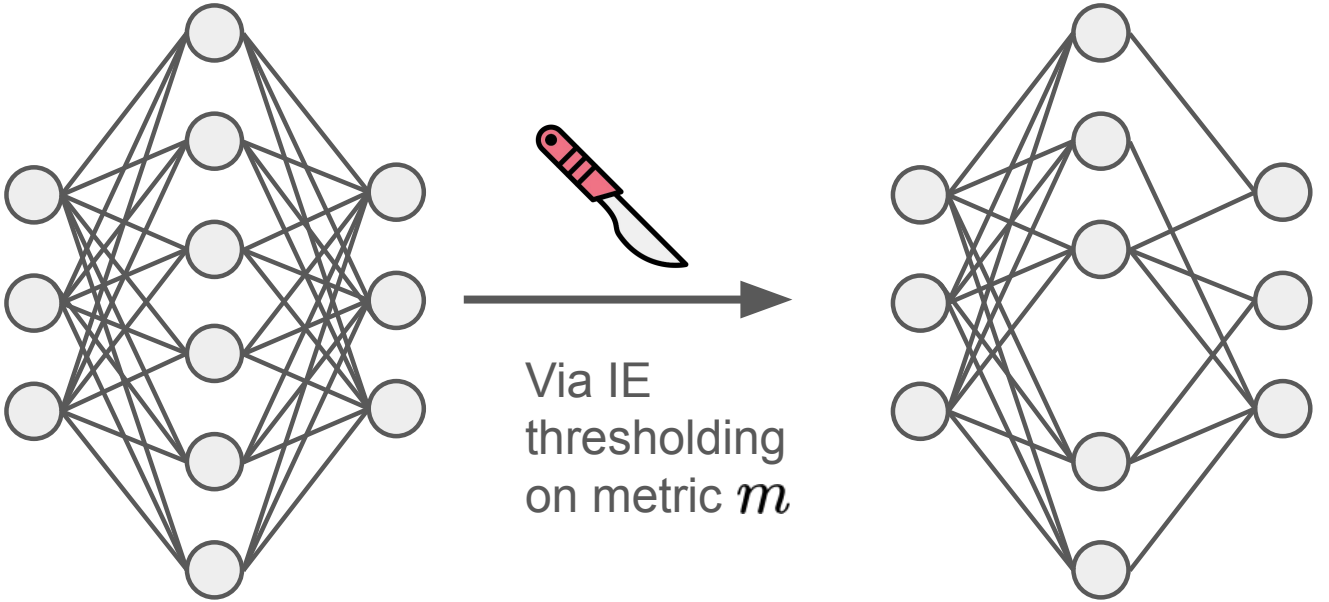


Sparse Feature Circuits

Detailed figure of Injected SAE into the model computational graph



Feature Ablation from SAE



$$\bar{x} = SAE(\bar{x}) + \epsilon$$

$$\bar{x} \approx SAE'(\bar{x}) + \epsilon$$

Sparse Human-Interpretable Feature Trimming

Method for removing unwanted model features from a LM classifier

Used to force Bias in Bios study model to learn occupational information rather than gender bias

1. Using metric m(x) = -log C(y | x), compute feature circuits that explain C's accuracy
2. Manually inspect those features for task-related ones
3. Ablate/ remove ones that aren't related
4. (Optional) further fine tune Classifier C

Case Study—Bias in Bios (BiB)

Bias in Bios is a dataset with spurious features such as gender with the primary task of classifying professions based off professional biography.

Training Dataset is unbalanced
Evaluation Dataset is balanced

Key Note: The dataset is purposely designed to misalign models

He is also the project lead of and major contributor to the open source assembler/simulator "EASy68K." He earned a master's degree in computer…	21	0
She is able to assess, diagnose and treat minor illness conditions and exacerbations of some long term conditions. Her qualifications include…	13	1
Prior to law school, Brittni graduated magna cum laude from DePaul University in 2011 with her Bachelor's Degree in Psychology and Spanish. I…	2	1

Bias in Bios Dataset [De-arteaga et al 2019]

Case Study—Bias in Bios (BiB)

Profession Accuracy = Average accuracy of all predicted professions
Gender Accuracy = Maximum percentage of male/females in a single profession
Worst Group = Worse Performance on single profession
Oracle = Trained directly on Balanced Dataset

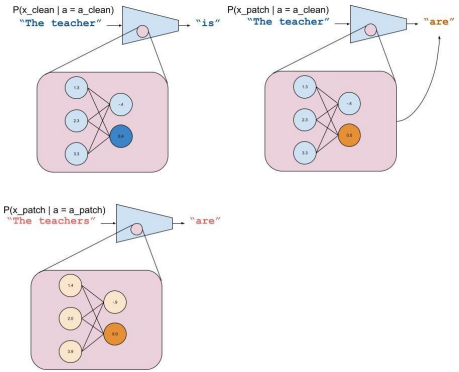
	Pythia-70M			Gemma-2-2B		
Method	↑Profession	↓Gender	↑Worst group	↑Profession	↓Gender	↑Worst group
Original	61.9	87.4	24.4	67.7	81.9	18.2
CBP	83.3	60.1	67.7	90.2	**50.1**	86.7
Random	61.8	87.5	24.4	67.3	82.3	18.0
SHIFT	88.5	54.0	76.0	76.0	51.5	50.0
SHIFT + retrain	**93.1**	**52.0**	**89.0**	**95.0**	52.4	**92.9**
Neuron skyline	75.5	73.2	41.5	65.1	84.3	5.6
Feature skyline	88.5	54.3	62.9	80.8	53.7	56.7
Oracle	93.0	49.4	91.9	95.0	50.6	93.1

Figure Adapated from [Marks et al 2025]

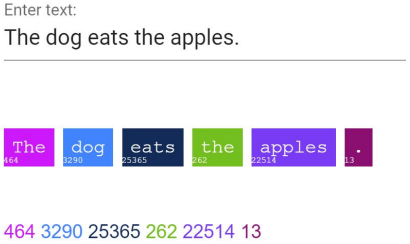
Automated Feature Discovery

Tasks and Contribution

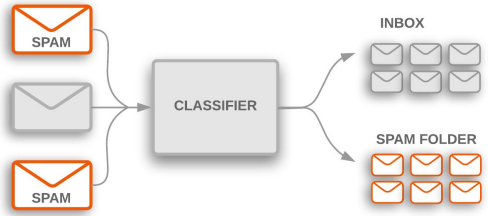
They showcase created Sparse Feature Circuits could be created for several different NLP tasks:



Subject verb agreement



Next token prediction



Classification

Research Contributions

- Development of the SHIFT technique for removing LM classifier sensitivity
- Fully unsupervised pipeline for computing thousands of feature circuits automatically
- Small feature circuits explain a large proportion of model behavior: the majority of performance in Pythia-70M,

SHIFT Limitations

SAE Limits

The paper requires many SAE's for SHIFT

Massive Upfront cost of training the SAE's

Model components not captured by SAEs will remain uninterpretable

Interpretability Limitations

Difficult to use paper methods without some downstream task

Feature labeling is a qualitative task and can change from annotator to annotator

Recall: Priming Questions

- Is it possible to fundamentally change how a model 'reasons' about a concept without showing it a single new piece of training data?
- If we 'cut out' a specific thought from a model's brain, do we lose the 'intelligence' surrounding it, or can we truly isolate a single unintended signal?
- If Fine-tuning (RLHF, SFT, RLVR, RLAIF) can teach a model to act a certain way towards us, how do we know if the bias is actually gone—or if the model has just learned to hide natural propensities?

Solutions to the Questions

- Yes, using Sparse Feature Circuits we are able to ablate features from the models internal representation and fundamentally alter the models computational graph.
- No, cutting out components of a model internal representation does not result in a loss of capability on specific tasks and can actually boost performance.
- Maybe, because fine tuning still preserves the connections between neurons and does not ablate them strictly

Thank You

Ethan Nguyen
nguyen@e-10.net

Matt Majeske
mmajeske@charlotte.edu

Published as a conference paper at ICLR 2025

SPARSE FEATURE CIRCUITS: DISCOVERING AND EDITING INTERPRETABLE CAUSAL GRAPHS IN LANGUAGE MODELS

Samuel Marks* Northeastern University
Can Rager Independent
Eric J. Michaud MIT
Yonatan Belinkov Technion – IIT
David Bau Northeastern University
Aaron Mueller* Northeastern University

ABSTRACT

We introduce methods for discovering and applying **sparse feature circuits**. These are causally implicated subnetworks of human-interpretable features for explaining language model behaviors. Circuits identified in prior work consist of polysemantic and difficult-to-interpret units like attention heads or neurons, rendering them unsuitable for many downstream applications. In contrast, sparse feature circuits enable detailed understanding of unanticipated mechanisms in neural networks. Because they are based on fine-grained units, sparse feature circuits are useful for downstream tasks: We introduce SHIFT, where we improve the generalization of a classifier by ablating features that a human judges to be task-irrelevant. Finally, we demonstrate an entirely unsupervised and scalable interpretability pipeline by discovering thousands of sparse feature circuits for automatically discovered model behaviors.

1 INTRODUCTION

The key challenge of interpretability research is to scalably explain the many unanticipated behaviors of neural networks (NNs). Much recent work explains NN behaviors in terms of coarse-grained model components, for example by implicating certain induction heads in in-context learning (Olsson et al., 2022) or MLP modules in factual recall (Meng et al., 2022; Geva et al., 2023; Nanda et al., 2023, inter alia). However, such components are generally polysemantic (Elhage et al., 2022) and hard to interpret, making it difficult to apply mechanistic insights to downstream applications. On the other hand, prior methods for analyzing behaviors in terms of fine-grained units (Kim et al., 2018; Belinkov, 2022; Geiger et al., 2023; Zou et al., 2023) attempt to fit model internals to researcher-specified mechanistic hypotheses using researcher-curated data. These approaches are not well-suited to the many cases where researchers cannot anticipate ahead of time how models internally implement their surprising behaviors.

We propose to explain model behaviors using fine-grained components that play narrow, interpretable roles. Doing so requires us to address two challenges: First, we must identify an appropriate fine-grained unit of analysis, since obvious choices like neurons[1] are rarely interpretable, and units discovered via supervised methods like linear probing require pre-existing hypotheses (Mueller et al., 2024). Second, we must address the scalability problem posed by searching for causal circuits over a large number of fine-grained units.

We leverage recent progress in dictionary learning for NN interpretability (Bricken et al., 2023; Cunningham et al., 2024) to tackle the first challenge. Namely, we use **sparse autoencoders** (SAEs) to identify directions in LM latent spaces which represent human-interpretable concepts. Then, to address the scalability challenge, we employ linear approximations (Kramár et al., 2024; Nanda

*Correspondence to s.marks@northeastern.edu and a.mueller@northeastern.edu.
[1]We use "neuron" to refer to a basis-aligned direction in an LM's latent space (not necessarily preceded by a nonlinearity).

1

Mark, Can, & et al's Paper

Presentation