Use Sparse Autoencoders to Discover Unknown Concepts, Not to Act on Known Concepts

Anonymous Author(s)
Affiliation
Address
email

Abstract

While sparse autoencoders (SAEs) have generated significant excitement, a series of negative results have added to skepticism about their usefulness. Here, we establish a conceptual distinction that reconciles competing narratives surrounding SAEs. We argue that while SAEs are less effective tools for *acting on known concepts*, SAEs are powerful tools for *discovering unknown concepts*. This distinction cleanly separates existing negative and positive results, and suggests several classes of SAE applications. Specifically, we outline use cases for SAEs in (i) text as data, (ii) bridging prediction and explanation in ML-based science, and (iii) ML interpretability, explainability, fairness, and auditing.

1 Introduction

Sparse autoencoders (SAEs) have been a popular topic in interpretability research, showing impressive capabilities for identifying interpretable directions in the text representations underlying language models [Cunningham et al., 2023, Templeton et al., 2024]. For example, an Anthropic paper found a "Golden Gate Bridge" direction, which could be manipulated to make a chatbot that would always incorporate the Golden Gate Bridge into responses.[1]

However, two recent papers showed that SAEs fail to outperform simple baselines in large-scale evaluations on concept detection (probing) and model steering [Kantamneni et al., 2025, Wu et al., 2025]. These results have led to pessimism about the usefulness of SAEs. For example, in response to this research, the mechanistic interpretability team at Google DeepMind announced that they would deprioritize research into SAEs.[2] Nonetheless, there continues to be optimism about new applications of SAEs: including in hypothesis generation [Movva et al., 2025] as well in the "biology" of LLMs [Lindsey et al., 2025].

How can we square continued interest in SAEs with thorough evaluations demonstrating negative results? Are new attempts to use SAEs misguided? Or is there something missing in our understanding of the negative results? This position paper reconciles conflicting narratives surrounding SAEs by making a conceptual distinction. **Our position is that SAEs—while ineffective at *acting on known concepts*—are powerful tools for *discovering unknown concepts*:**

Consider the tasks where *negative* results have been shown. Concept detection involves detecting a prespecified concept ("Does this text mention dogs?"). Model steering involves steering a model to exhibit a specified concept ("Make outputs less sycophantic."). Another negative result involves concept unlearning ("Unlearn knowledge about concepts related to biosecurity.") [Farrell et al., 2024] In these tasks, concepts are *inputs*—known beforehand.

[1] https://www.anthropic.com/news/golden-gate-claude

[2] https://deepmindsafetyresearch.medium.com/negative-results-for-sparse-autoencoders-on-downstream-tasks-and-deprioritising-sae-research-6cadcfc125b9

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Now consider the tasks where *positive* results have been shown. Hypothesis generation involves finding concepts that predict a target variable ("What concepts predict engagement of news headlines?"). Biology of LLMs involves finding concepts that LLMs represent when generating text ("What concepts does an LLM represent when doing addition?"). In both tasks, concepts are *outputs*—unknown beforehand. (In Table 1, we provide example concepts.)

So while SAEs have been shown to underperform baselines when acting on knowns, overgeneralizing can result in missing the potential of SAEs as tools for discovering unknown concepts. By enumerating concepts in an unsupervised manner, SAEs allow for the discovery of concepts that fit desired criteria. We outline how SAEs—as a tool for generating unknown concepts—can be used to advance research in (i) applications of text as data, (ii) the role of prediction and explanation in ML-based science, and (iii) ML interpretability, explainability, fairness, and auditing.

Paper structure. Section 2 serves as a primer on SAEs (which can be readily skipped by readers familiar with SAEs). Section 3 shows that negative SAE results pertain to tasks that act on known concepts. Section 4 surveys recent positive results, showing that these papers use SAEs to discover unknown concepts. Section 5 then explores use cases for SAEs in different research areas.

2 An SAE Primer

We offer a brief primer on the SAE architecture, their history, and why and how they are now being used to interpret language models. (Readers familiar with SAEs may skip to Section 3.)

Early work on autoencoders. Autoencoders are unsupervised neural networks that learn to reconstruct high-dimensional inputs via a series of learned transformations. For a D-dimensional input $\mathbf{x}$, an autoencoder computes:

$$\mathbf{z} = \text{encoder}(\mathbf{x}), \tag{1}$$

$$\hat{\mathbf{x}} = \text{decoder}(\mathbf{z}), \tag{2}$$

where $\text{encoder}(\cdot), \text{decoder}(\cdot)$ are arbitrary neural networks, $\mathbf{z}$ is the *latent* feature representation, and $\hat{\mathbf{x}}$ is the *reconstruction*. The autoencoder is trained with a mean squared error reconstruction loss,

$$\mathcal{L} = ||\hat{\mathbf{x}} - \mathbf{x}||_2^2. \tag{3}$$

One classic application of autoencoders is compression: by restricting the latent representation $\mathbf{z}$ to a dimension size $M \ll D$, the autoencoder learns a compressed representation in $\mathbf{z}$ which can be used to approximate $\mathbf{x}$ [Hinton and Salakhutdinov, 2006]. In this setting, $\mathbf{z}$ functions similarly to an M-dimensional principal component analysis of $\mathbf{x}$,[3] in that we wish to explain as much variance as possible in the distribution of $\mathbf{x}$ using only M dimensions.

Sparse autoencoders. Sparse autoencoders (SAEs) perform the same reconstruction task, but leverage a different intuition. In an SAE, M can be *larger* than D, but each individual $\mathbf{z}$ is forced to be sparse—that is, only a small number of its dimensions can be nonzero. This design is motivated by the idea that while an entire dataset may span many possible concepts (e.g., all text on the Internet, or all images in ImageNet), a single datapoint (a sentence or image) often contains very few. Empirically, using this design results in dimensions of $\mathbf{z}$ which correspond to useful concepts. For example, early work on SAEs trains directly on input images $\mathbf{x}$ (e.g., from MNIST), and the features learned by $\mathbf{z}$ correspond to interpretable concepts like edges [Coates and Ng, 2011, Makhzani and Frey, 2014].

To improve clarity, we define *features* and *concepts*:

- A **feature** is one of many numerical values used to represent an input. In a neural network, a feature is a single **dimension** of a layer's output vector; in other words, it is an **activation** computed by a **neuron**. We use the terms feature, activation, neuron, and dimension interchangeably depending on context.
- A **concept** is a qualitative characteristic that may or may not be present in a given input. For our purposes, we operationalize concepts via natural language descriptions.
- An **interpretable feature**, then, is a feature whose values correspond to the presence of absence of a single concept.

[3] If the encoder and decoder are linear, PCA minimizes the reconstruction loss [Baldi and Hornik, 1989].

Concepts	Example Texts
News headlines [Movva et al., 2025]	
Protests or actions of dissent	"Sometimes Silence Is The Best Form Of Protest"
"How to / what to do" questions or instructions	"Why are People In Mexico Taking To The Streets?"
Economic inequality	"It Would Be Revolting To Not Stand Up For What You Believe ..."
Memory or remembering	"A massive, global protest is going down today. You should know why."
Direct requests or demands	"This May Be the Most Important Battle Of Our Times ..."
Drugs or drug-related topics	"As riots broke out ... this group of Baltimore clergy marched in peaceful protest"
Gov't policies related to democracy, citizen rights	"The Internet is Important To Protest Movements, But It's Not Always Used to HELP Them."
Climate change or global warming	
Hollywood or the film industry	
Cats or cat-related topics	
Congressional Speeches [Movva et al., 2025]	
Tax cuts or benefits for the wealthy	"Doing nothing is the worst thing Congress can do ..."
The national debt or debt ceiling	"We need to stop the rhetoric and take action ..."
North Dakota or its communities	"... for evil to triumph it is only necessary that good men do nothing."
Criticizes inaction or lack of progress by Congress	"... it seems to me that one way to raise it would be to do something"
Tax relief or royalty relief	"There is no action whatsoever in this bill ..."
Postal Service or postal reform	"They simply do not want to do it. But what they want to do now is just throw some additional money at it to kind of kick the can ..."
High-ranking military officers	"... we are not doing anything but saying we are going to go right ..."
A person named Katie or Kathryn	
Price-gouging or energy market	
Phrases emphasizing negation or absence	
General Text Corpus [Lindsey et al., 2025]	
Visual deficits	"... and Byzantine art was mainly found in the Roman Empire"
Something that ends in "it"	"... clashes between the Blues and Greens in Constantinople ..."
Answering difficult questions/ sensitive questions	"... Eastern Roman Empire which is what we call Byzantium ..."
Meningitis symptoms	"... Egypt and Byzantine art was mainly found in the Roman Empire ..."
Everything's bigger in Texas	"... Eastern Roman Empire, also known as the Byzantine Empire ..."
Two-digit numbers in the 10-20 range	"... la hiérarchie qui existaient sous l'empire d'Orient..."
Rabbit	"... reconoció formalmente al emperador romano de Oriente ..."
Byzantine Empire	
Can't answer	
Dangers of Bleach and Ammonia	

Table 1: SAE neurons explained via autointerpretation, and texts that activate them [Movva et al., 2025, Lindsey et al., 2025]. **Left:** Examples of concepts learned from SAEs trained on different datasets; **Right:** Examples of texts that activate the corresponding SAE neuron. Concepts interpretably describe the underlying data distribution of texts.

Mathematical formulation of SAEs. One formulation of the sparse autoencoder follows the usual autoencoder forward pass, but adds an L_1 penalty on $\mathbf{z}$ to the loss function [Coates and Ng, 2011]:

$$\mathcal{L} = ||\hat{\mathbf{x}} - \mathbf{x}||_2^2 + \lambda||\mathbf{z}||_1. \tag{4}$$

A larger λ encourages more zero elements in $\mathbf{z}$. Another approach explicitly applies a TopK function to the encoder that zeroes out all but the k-largest activations in $\mathbf{z}$ [Makhzani and Frey, 2014], where $k \ll M$. These top-k SAEs use a vanilla reconstruction loss. With a single layer each for the encoder and decoder, the full forward pass is given by:

$$\begin{aligned}\mathbf{z} &= \mathrm{ReLU}(\mathrm{TopK}(W_{\mathrm{enc}}(\mathbf{x} - \mathbf{b}_{\mathrm{pre}}) + \mathbf{b}_{\mathrm{enc}})),\\ \hat{\mathbf{x}} &= W_{\mathrm{dec}}\mathbf{z} + \mathbf{b}_{\mathrm{dec}},\end{aligned}$$

where $\mathbf{b}_{\mathrm{pre}} \in \mathbb{R}^D, W_{\mathrm{enc}} \in \mathbb{R}^{M\times D}, \mathbf{b}_{\mathrm{enc}} \in \mathbb{R}^M, W_{\mathrm{dec}} \in \mathbb{R}^{D\times M}, \mathbf{b}_{\mathrm{dec}} \in \mathbb{R}^D$, and TopK sets all activations except the top k to zero.

Single layer top-k SAEs have emerged as a common architecture in recent work, with slight variations like an auxiliary loss or nested losses to mitigate issues like dead neurons and feature absorption [Gao et al., 2024, Bussmann et al., 2025]. Some work has replaced SAEs with sparse *transcoders*, which use layer ℓ_i to construct the output of a later layer ℓ_j [Paulo et al., 2025, Lindsey et al., 2025]. For convenience, **we refer to all of these closely-related methods under the "SAE" umbrella**, while noting that the specifics of the optimal sparse coding architecture are likely to shift.

Applying SAEs to interpret language models. The recent wave of SAE research aims to interpret the representations learned by large language models. The motivation for this line of work is to understand the units and computations an LM uses to map an input to an output. Before SAEs, a plethora of works over the last decade on *probing* language models have shown that LM token representations contain rich semantic information [Belinkov, 2022]. Concepts like a word's part-of-speech or pronoun coreferences are a linear transformation away from the word's representation [Liu et al., 2019]. Given this richness, a natural question is whether we can identify all of the concepts a language model encodes and the model components that encode them. A starting point is to interpret a single neuron [Elhage et al., 2022]. Unfortunately, individual neurons are hard to describe in a human-interpretable way. Neurons tend to capture a complex combination of concepts, and this *polysemanticity* appears to be a fundamental property of neural networks [Elhage et al., 2022].

This convergence of findings—that language model representations encode numerous valuable concepts, but studying individual neurons does not reveal them—explains recent excitement for sparse autoencoders. Unlike LM neurons, SAEs produce *monosemantic* neurons that *can* be explained by a single concept [Cunningham et al., 2023, Bricken et al., 2023]. SAEs are trained on an LM's representations $\mathbf{x}$ of individual tokens, resulting in latent representations $\mathbf{z}$. To interpret a particular feature dimension i in $\mathbf{z}$, we can examine tokens (and their surrounding context) that produce large values of $\mathbf{z}[i]$. Initial work reports that after training on the representations from a small, one-layer LM, the SAE features $\mathbf{z}[i]$ fire on succinct concepts, like "*Arabic text*" or "*citations in scientific papers*" [Cunningham et al., 2023, Bricken et al., 2023]. Follow-up work demonstrates that SAEs continue to learn monosemantic features when applied to representations from state-of-the-art LLMs [Templeton et al., 2024, Gao et al., 2024]. SAEs also produce interesting features when trained on text embeddings of entire sentences or documents [O'Neill et al., 2024, Movva et al., 2025]. In Table 1, we provide examples of concepts learned on both specific text datasets (news headlines and Congressional speeches) as well as generic text datasets.

Automatically interpreting neurons with language models. While SAEs produce neurons in $\mathbf{z}$ that are theoretically interpretable, the task of actually producing a mapping from neurons to concepts is a separate one. Because there are many neurons to explain, prior work has focused on automatically generating explanations of SAE neurons[4] (Templeton et al. [2024], O'Neill et al. [2024], *inter alia*). To interpret a neuron i, a basic approach is to prompt a language model with texts that have a high value of $\mathbf{z}[i]$ against those with a low value, and ask it to identify the shared concept in the high-valued texts. To evaluate the quality of the resulting concept description, one can use an LM to annotate texts for the presence of the concept, and measure agreement between the concept annotations and the true

[4]Many key works on neuron explanation interpret neurons in language or vision models directly, without SAEs [Bau et al., 2017, Hernandez et al., 2022, Bills et al., 2023, Choi et al., 2024]. The value proposition of SAEs is that, relative to the original model's neurons, SAE features are easier to explain with high fidelity.

Neg. Results: Acting on Known Concepts	Pos. Results: Discovering Unknown Concepts
Concept detection [Wu et al., 2025, Kantamneni et al., 2025]	**Hypothesis Generation** [Movva et al., 2025]
Is the following name a basketball player?	What concepts predict engagement on news headlines?
Is the following entity in New York City?	What concepts predict partisanship in Congressional speeches?
Model steering [Wu et al., 2025]	**Biology of LLMs** [Lindsey et al., 2025]
Make the LLM output more sycophantic.	What concepts does an LLM represent after writing the first line of a poem?
Make the LLM output discuss the Golden Gate Bridge.	What concepts does an LLM represent when performing addition?

Table 2: Negative SAE results *act on known concepts* whereas positive SAE results focus on *discovering unknown concepts*.

neuron activations. This framework gives us a quantitative measure for neuron interpretability: how well does a natural language explanation predict the neuron's activations?

There are ongoing debates about how best to sample high- and low-valued texts both during explanation and during scoring in order to produce the "best" explanation [Bills et al., 2023, Gao et al., 2024, Movva et al., 2025]. For example, even a generic explanation may distinguish the top-valued texts from random ones, but an overly specific one may miss medium-valued activations. Another proposal to score explanations asks an LM to generate text which contains the concept, and then measures whether the generated text indeed has a high activation [Juang et al., 2024]. However, this does not resolve issues around explanation specificity. More broadly, there is no consensus for automatic neuron explanation; indeed, the choice should be grounded in the task the concepts are being used for.

3 Negative results: Acting on Known Concepts

We now survey recent negative results about SAEs, with the goal of showing that the tasks considered fall under the category of acting on known concepts. This is to be contrasted with tasks that involve discovering unknown concepts, on which positive results have been shown (Section 4).

Two recent papers conduct large-scale evaluations of SAEs [Kantamneni et al., 2025, Wu et al., 2025]. A key finding of these papers is that SAEs underperform simple baseline methods (such as logistic regression or naive prompting). We claim that these evaluations are limited to tasks involving **acting on known concepts**. Indeed, the tasks that are studied are:

1. Concept detection [Kantamneni et al., 2025, Wu et al., 2025]: Identifying whether a given concept appears in a text.
2. Model steering [Wu et al., 2025]: Steering the outputs of a language model to contain a concept.

These are important, widely-studied problems, and understanding how SAEs perform on them is clarifying. Notice, however, that these tasks each involve first prespecifying a concept and then acting upon it. In other words, concepts are inputs in these tasks. We now summarize these papers' findings in greater detail.

Concept detection. Kantamneni et al. [2025] curate 113 binary classification tasks on text data, which they use to evaluate concept detection accuracy. For example, one task is to determine if a given name corresponds to a basketball player. Another task is to determine if a tweet conveys happy sentiment. They train *probing classifiers*: on each dataset, they fit a logistic regression to predict concept presence using `Gemma-2-9B`'s representations of the final tokens in each text as input[5]. They compare this to a logistic regression trained on the representations from a `Gemma-2-9B`

[5]Besides logistic regression, they also include PCA regression, nearest neighbors, XGBoost, and MLP.

SAE. They further examine class imbalance, data scarcity, and label noise. In each setting, using the SAE representation does not add predictive power compared to probing directly from the LM representation.

Wu et al. [2025] follow a similar approach. Starting from a list of 500 concepts, for each concept, they generate synthetic texts that either do or do not contain the concept. In addition to logistic regression using `Gemma-2-2B` representations, they train several other representation-based concept detection methods. They also include methods that do not use representations at all, such as prompting an LLM to identify whether the concept is present in the text, as well as bag-of-words. Four such baselines, including logistic regression and prompting, outperform the SAE.

Model steering. Wu et al. [2025] also study model steering. Given a user prompt and a concept, like "where should I visit today?" and "Golden Gate Bridge," they evaluate whether the model can generate a response that is fluent, relates to the prompt, and includes the concept. An LLM judge scores each attribute. To steer with an SAE, they identify the SAE feature that is most predictive of the concept's presence, and they generate a response after increasing the value of this feature. Non-SAE methods include editing activations with a steering vector [Marks and Tegmark, 2024], finetuning the language model on responses containing the concept, or simply prompting it to include the concept in its response. Prompting and finetuning both outperform SAE-based steering.

Why aren't SAEs useful here? We speculate on why SAEs underperform baselines on these tasks. For concept detection, recall that SAEs are trained to reconstruct the LM token representations. A reconstruction encodes strictly less information about a token than the the original LM representation. It follows that, compared to the original representation, there is less information available in the SAE representation to predict the presence of a concept. For model steering, prompting performs well because LLMs are finetuned to be adept at instruction-following, and including a concept in a response falls well within this paradigm. The empirical results from both papers underscore an intuition that, more generally, there are many natural methods besides SAEs to act on known concepts. However, these baselines are less equipped to perform another simple task: enumerate a list of candidate concepts. This, as we show in the next section, forms the basis for tasks on which SAEs have a comparative advantage.

4 Positive results: Discovering Unknown Concepts

We now describe two positive results using SAEs[6] [Movva et al., 2025, Lindsey et al., 2025], which focus on the following tasks:

1. Hypothesis generation [Movva et al., 2025]: Identifying open-ended natural language concepts that predict a target variable.
2. Explaining language model outputs ("Biology of LLMs") [Lindsey et al., 2025]: Describing the concepts a language model uses to perform various tasks (e.g., poem completion or addition).

We claim that these tasks are examples of **discovering unknown concepts**. To explain this, we summarize their findings in greater detail.

Hypothesis generation. Movva et al. [2025] study tasks where a large dataset of texts is annotated with a target variable, and the goal is to understand what concepts in the text predict the target. For example, one such dataset consists of news headlines and numerical engagement levels. While a traditional analysis of such a dataset may be hypothesis-driven (e.g., Robertson et al. [2023] study how negativity affects engagement), here the task requires automatically extracting concepts with no prior specification.

They (1) train an SAE on dense text embeddings; (2) select SAE features that predict the target; and (3) run autointerpretation to interpret the selected features, which become hypotheses (i.e., "*headlines that contain {concept} receive more engagement*"). They find that the resulting hypotheses outperform those generated without an SAE, either by skipping step 1 and selecting features directly from text

[6]Note that Lindsey et al. [2025] use sparse transcoders, a slight variation on SAEs (see note in §2).

Research area	Research problem (using SAEs to discover unknown concepts)
Text as data	How has language about immigration changed over time in Congressional speeches? [Card et al., 2022] What symptoms (recorded in medical records) predict clinical outcomes? [Huang et al., 2019] What information from court hearings do judges use when making bail decisions? [Zhang, 2024]
Explanation vs. prediction in ML-based science	What features explain the difference in accuracy between predictive models and theory-grounded models? [Fudenberg et al., 2022] Are ML models using illegitimate features (in the context of making a scientific claim)? [Kapoor and Narayanan, 2023]
ML interpretability	Finding natural language concepts that can be used to build an inherently interpretable model. [Rudin, 2019]
ML explainability	Finding natural language concepts that explain a model's predictions. [Lakkaraju et al., 2019]
ML fairness/bias	In what ways do LLMs stereotype different demographic groups? [Lucy and Bamman, 2021]
ML auditing	What features are high-stakes LLM-based decision tools using? [Gaebler et al., 2024]

Table 3: Example research problems that can use SAEs to discover unknown concepts.

embeddings, or by using a different pipeline altogether (like topic modeling or n-grams). The method quantitatively outperforms existing methods for hypothesis generation on text data—generating more statistically significant hypotheses. In hypothesis generation, the goal is explicitly to discover unknown concepts.

Explaining language model outputs. Lindsey et al. [2025] explain how language models generate text that completes a task. For example, given a prompt "A rhyming couplet: He saw a carrot and had to grab it," the LM generates the next line "His hunger was like a starving rabbit." Does the model generate the first part of the line and then "improvise" a word that rhymes, does it "plan" the rhyming word and tee it up, or something else? They find that, immediately after the first line, there are active neurons corresponding to the words "*rabbit*" and "*habit*" that result from a neuron "*words rhyming with 'it'*." They perform further interventions to confirm this "planning" mechanism. In another case, they look at how a model computes "36+59" in natural language. They find active neurons for "*units digit 5*," (resulting from a neuron for "*units digit 6 + units digit 9*") and "*addition problems of ∼40 plus ∼50*," which combine to produce "95." These specific routes of task completion are difficult to forecast, underscoring how this analysis requires discovering unknown concepts.

Why are SAEs useful here? Because SAEs are able to generate highly-interpretable features while maintaining the expressivity of underlying text representations, it is possible to identify concepts that satisfy a property: in the case of hypothesis generation, to find concepts that predict a target variable, and in the case of explaning language model behaviors, to find concepts that are active when completing a task. Precise concepts are important. If the *rabbit* neuron instead fired on all animals, it would be difficult to answer whether the model improvises or plans rhymes.

Also note that after generating concepts using SAEs, it is possible to computationally validate whether the concepts satisfy the desired property. That is, it is possible to evaluate whether a hypothesized headline concept indeed correlates with engagement, or whether a hypothesized LLM addition feature is active during addition. Because of this falsifiability, even if an SAE feature is unreliable (e.g., not all headlines which pose a question activate the question feature), it is possible to catch these issues downstream.

By enumerating a set of precise concepts that express the variation in text data, it is possible to systematically discover concepts that satisfy a desired property.

5 Use Cases for SAEs

Having conceptualized where SAEs are useful (discovering unknown concepts), we outline research areas where such a capability can be useful. In particular, while initial excitement about SAEs was shared primarily by researchers in mechanistic interpretability, we believe that clarifying the comparative advantage of SAEs reveals a significantly broader set of uses. The use cases we outline focus on the ability of SAEs to discover unknown concepts.

We first articulate why SAEs are a promising tool for *text as data*. SAEs have the potential to significantly improve upon methods that currently use keyword frequency or topic models. We then consider how SAEs can be used to bridge the prediction-explanation gap in ML-based science. Finally, we discuss how SAEs are an important tool for ML researchers in interpretability, explainability, fairness, and auditing to build upon and refine—especially insofar as these fields deal with models that use unstructured text data as both input and output. We summarize these potential use cases of SAEs for different research problems in Table 3.

Text as data. A wide variety of disciplines (e.g., sociology, economics, healthcare) have sought to leverage large text datasets. This has led to prominent work developing and applying methods for "text as data" [Grimmer, 2010, Gentzkow et al., 2019].

These methods often attempt to discover interpretable patterns in text data—for example, quantifying changes in the language used to discuss immigrants, or identifying features of clinical notes that predict health outcomes. Existing methods automate these tasks through simple text features such as keywords or n-grams, or through topic models. These methods are limited by the expressivity of these features: topic models and keywords do not precisely capture the range of concepts present in text.

At the same time, while text embeddings better capture the information present in text, they are uninterpretable. SAEs, by learning interpretable features from text embeddings, can be used to answer the same questions that previous keyword or topic model methods are used for—i.e., discovering concepts that reveal patterns in text—but potentially with higher quality.

ML-based science: bridging prediction and explanation. There are many settings in which text data have been shown to enable much greater predictive accuracy than existing human-specified features. While developing methods to quantify or improve predictive accuracy may be of independent interest, a growing line of work has suggested the need to bridge the gap between prediction and explanation [Hofman et al., 2017, 2021]. Traditionally, scientific disciplines have sought to *explain* phenomena, rather than only predict outcomes. For example, Fudenberg et al. [2022] and Ludwig and Mullainathan [2024] each show gaps between predictive accuracy of ML models that take in all available features and models that take in existing human-specified features. This gap suggests that existing theories are incomplete, leading to work that has sought to build automated approaches for closing this gap: discovering interpretable features that are predictive. SAEs are a promising tool for this task [Movva et al., 2025]. SAEs can close the prediction-explanation gap by converting black box representations into interpretable representations. These interpretable representations both capture much of the predictive power of the black box representations, while also enabling us to make predictions in terms of natural language concepts.

Other work has demonstrated that strong predictive performance can be misleading in ML-based science applications, underscoring the need for explanation [Kapoor et al., 2024, Messeri and Crockett, 2024, Del Giudice et al., 2024, Shmueli, 2010]. For example, ML models with high accuracy may use illegitimate or spurious features. We provide a concrete example from the results described in Movva et al. [2025]. In a dataset of Congressional speeches, several of the most predictive features of partisanship are procedural in nature, such as calling a session to order. While predictive accuracy in predicting partisanship could be used to measure substantive difference in partisan rhetoric, it is important to understand how much these procedural features contribute to the predictions. Similarly, even if text data in medical documents are highly predictive of patient outcomes, it is important to discover spurious features [Ross, 2021, Chiavegatto Filho et al., 2021]. In unstructured text data, discovering these illegitimate features can be difficult. SAEs provide one way of discovering these features. This capability extends past methods that *prespecify* concepts to be used for prediction with unstructured data [Koh et al., 2020].

For ML interpretability/explainability/fairness/auditing. Each of these areas aim to understand and build models with desiderata beyond accuracy in mind. Here, we see significant opportunity for SAEs. For example, SAEs can be used to identify natural language concepts that can explain black box model behavior [Lakkaraju et al., 2019]. Then, by identifying the concepts that are used, it is possible to build models that are inherently interpretable [Rudin, 2019], and that incorporate only features that we want (e.g., that are considered fair, avoid spurious correlations, etc).

Whereas existing work documents how demographic information affect LLM-based decision-making—e.g., in hiring [Gaebler et al., 2024], it is possible to use SAEs to uncover a wider range of features that may affect the LLM-generated decisions. Furthermore, as LLM outputs are themselves often unstructured, SAEs can also be used to discover patterns in LLM outputs as a function of inputs, expanding the toolkit of researchers auditing models.

6 Conclusion

In this position paper, we argued that successful uses of SAEs involve discovering new concepts, while unsuccessful uses of SAEs involve acting on known concepts. We showed that negative results have used SAEs to act on known concepts—e.g., on tasks such as concept detection and model steering. Meanwhile, positive results using SAEs—including hypothesis generation and biology-of-LLMs—have aimed to discover unknown concepts using SAEs. Having clarified where SAEs show promise, we then outlined potential applications of SAEs.

References

P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, Jan. 1989. ISSN 0893-6080.

D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network Dissection: Quantifying Interpretability of Deep Visual Representations, Apr. 2017.

Y. Belinkov. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, 48(1):207–219, Apr. 2022. ISSN 0891-2017.

S. Bills, N. Cammarata, D. Mossing, H. Tillman, L. Gao, G. Goh, I. Sutskever, J. Leike, J. Wu, and W. Saunders. Language models can explain neurons in language models. `https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html`, 2023.

T. Bricken, A. Templeton, J. Batson, B. Chen, A. Jermyn, T. Conerly, N. Turner, C. Anil, C. Denison, A. Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2, 2023.

B. Bussmann, N. Nabeshima, A. Karvonen, and N. Nanda. Learning Multi-Level Features with Matryoshka Sparse Autoencoders, Mar. 2025.

D. Card, S. Chang, C. Becker, J. Mendelsohn, R. Voigt, L. Boustan, R. Abramitzky, and D. Jurafsky. Computational analysis of 140 years of us political speeches reveals more positive but increasingly polarized framing of immigration. *Proceedings of the National Academy of Sciences*, 119(31): e2120510119, 2022.

A. Chiavegatto Filho, A. F. D. M. Batista, and H. G. Dos Santos. Data leakage in health outcomes prediction with machine learning. comment on "prediction of incident hypertension within the next year: prospective study using statewide electronic health records and machine learning". *Journal of medical Internet research*, 23(2):e10969, 2021.

D. Choi, V. Huang, K. Meng, D. D. Johnson, J. Steinhardt, and S. Schwettmann. Scaling automatic neuron description, October 2024. URL `https://transluce.org/neuron-descriptions`. Published online at Transluce AI.

A. Coates and A. Y. Ng. The importance of encoding versus training with sparse coding and vector quantization. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 921–928. Omnipress, June 2011. ISBN 978-1-4503-0619-5.

H. Cunningham, A. Ewart, L. Riggs, R. Huben, and L. Sharkey. Sparse Autoencoders Find Highly Interpretable Features in Language Models, Oct. 2023.

M. Del Giudice et al. The prediction-explanation fallacy: A pervasive problem in scientific applications of machine learning. *Methodology*, 20(1):22–46, 2024.

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, Sept. 2022. Comment: Also available at https://transformer-circuits.pub/2022/toy_model/index.html.

E. Farrell, Y.-T. Lau, and A. Conmy. Applying sparse autoencoders to unlearn knowledge in language models. *arXiv preprint arXiv:2410.19278*, 2024.

D. Fudenberg, J. Kleinberg, A. Liang, and S. Mullainathan. Measuring the completeness of economic models. *Journal of Political Economy*, 130(4):956–990, 2022.

J. D. Gaebler, S. Goel, A. Huq, and P. Tambe. Auditing large language models for race & gender disparities: Implications for artificial intelligence-based hiring. *Behavioral Science & Policy*, 10 (2):46–55, 2024.

L. Gao, T. D. la Tour, H. Tillman, G. Goh, R. Troll, A. Radford, I. Sutskever, J. Leike, and J. Wu. Scaling and evaluating sparse autoencoders, June 2024.

M. Gentzkow, B. Kelly, and M. Taddy. Text as data. *Journal of Economic Literature*, 57(3):535–574, 2019.

J. Grimmer. A bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases. *Political analysis*, 18(1):1–35, 2010.

E. Hernandez, S. Schwettmann, D. Bau, T. Bagashvili, A. Torralba, and J. Andreas. Natural Language Descriptions of Deep Visual Features, Apr. 2022.

G. E. Hinton and R. R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507, July 2006.

J. M. Hofman, A. Sharma, and D. J. Watts. Prediction and explanation in social systems. *Science*, 355(6324):486–488, 2017.

J. M. Hofman, D. J. Watts, S. Athey, F. Garip, T. L. Griffiths, J. Kleinberg, H. Margetts, S. Mullainathan, M. J. Salganik, S. Vazire, et al. Integrating explanation and prediction in computational social science. *Nature*, 595(7866):181–188, 2021.

K. Huang, J. Altosaar, and R. Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*, 2019.

C. Juang, G. Paulo, J. Drori, and N. Belrose. Open source automated interpretability for sparse autoencoder features. *EleutherAI Blog, July*, 30, 2024.

S. Kantamneni, J. Engels, S. Rajamanoharan, M. Tegmark, and N. Nanda. Are sparse autoencoders useful? a case study in sparse probing. *arXiv preprint arXiv:2502.16681*, 2025.

S. Kapoor and A. Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9), 2023.

S. Kapoor, E. M. Cantrell, K. Peng, T. H. Pham, C. A. Bail, O. E. Gundersen, J. M. Hofman, J. Hullman, M. A. Lones, M. M. Malik, et al. Reforms: Consensus-based recommendations for machine-learning-based science. *Science Advances*, 10(18):eadk3452, 2024.

P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020.

H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec. Faithful and customizable explanations of black box models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 131–138, 2019.

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. URL https://transformer-circuits.pub/2025/attribution-graphs/biology.html.

N. F. Liu, M. Gardner, Y. Belinkov, M. E. Peters, and N. A. Smith. Linguistic Knowledge and Transferability of Contextual Representations, Apr. 2019.

L. Lucy and D. Bamman. Gender and representation bias in GPT-3 generated stories. In N. Akoury, F. Brahman, S. Chaturvedi, E. Clark, M. Iyyer, and L. J. Martin, editors, *Proceedings of the Third Workshop on Narrative Understanding*, pages 48–55, Virtual, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.nuse-1.5. URL https://aclanthology.org/2021.nuse-1.5/.

J. Ludwig and S. Mullainathan. Machine learning as a tool for hypothesis generation. *The Quarterly Journal of Economics*, 139(2):751–827, 2024.

A. Makhzani and B. Frey. K-Sparse Autoencoders, Mar. 2014.

S. Marks and M. Tegmark. The Geometry of Truth: Emergent Linear Structure in Large Language Model Representations of True/False Datasets, Aug. 2024.

L. Messeri and M. Crockett. Artificial intelligence and illusions of understanding in scientific research. *Nature*, 627(8002):49–58, 2024.

R. Movva, K. Peng, N. Garg, J. Kleinberg, and E. Pierson. Sparse Autoencoders for Hypothesis Generation, Mar. 2025.

C. O'Neill, C. Ye, K. Iyer, and J. F. Wu. Disentangling Dense Embeddings with Sparse Autoencoders, Aug. 2024.

G. Paulo, S. Shabalin, and N. Belrose. Transcoders Beat Sparse Autoencoders for Interpretability, Feb. 2025.

C. E. Robertson, N. Pröllochs, K. Schwarzenegger, P. Pärnamets, J. J. Van Bavel, and S. Feuerriegel. Negativity drives online news consumption. *Nature Human Behaviour*, 7(5):812–822, May 2023. ISSN 2397-3374.

C. Ross. Epic's sepsis algorithm is going off the rails in the real world. the use of these variables may explain why. *Stat, September*, 27, 2021.

C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

G. Shmueli. To Explain or to Predict? *Statistical Science*, 25(3):289 – 310, 2010. doi: 10.1214/10-STS330. URL https://doi.org/10.1214/10-STS330.

A. Templeton, T. Conerly, J. Marcus, J. Lindsey, T. Bricken, B. Chen, A. Pearce, C. Citro, E. Ameisen, A. Jones, et al. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. transformer circuits thread, 2024.

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. *arXiv preprint arXiv:2501.17148*, 2025.

S. Zhang. Tipping the balance: Predictive algorithms and institutional decision-making in context. 2024.