Sparse Autoencoders for Hypothesis Generation

Rajiv Movva [*] [1] Kenny Peng [*] [2 3] Nikhil Garg [3] Jon Kleinberg [2] Emma Pierson [1]

Abstract

We describe HypotheSAEs, a general method to hypothesize interpretable relationships between text data (e.g., headlines) and a target variable (e.g., clicks). HypotheSAEs has three steps: (1) train a sparse autoencoder on text embeddings to produce interpretable features describing the data distribution, (2) select features that predict the target variable, and (3) generate a natural language interpretation of each feature (e.g., *mentions being surprised or shocked*) using an LLM. Each interpretation serves as a hypothesis about what predicts the target variable. Compared to baselines, our method better identifies reference hypotheses on synthetic datasets (at least +0.06 in F1) and produces more predictive hypotheses on real datasets (∼twice as many significant findings), despite requiring 1-2 orders of magnitude less compute than recent LLM-based methods. HypotheSAEs also produces novel discoveries on two well-studied tasks: explaining partisan differences in Congressional speeches and identifying drivers of engagement with online headlines.

1. Introduction

Large language models (LLMs) show promise as a tool for *hypothesis generation*. Discovering relationships between text data and a target variable is an important and fundamental task with diverse applications in economics, political science, sociology, medicine, and business (Grimmer, 2010; Rathje et al., 2021; Sun et al., 2022; Gentzkow & Shapiro, 2010; Monroe et al., 2009; Nelson, 2020; Ranard et al., 2016; Ting et al., 2017). What features of a restaurant review predict a low rating? What features of a social media post predict whether it will go viral? What features of a patient's clinical notes predict if they will develop cancer? Automated approaches to answering these questions have the potential to significantly expand and accelerate scientific discovery (Ludwig & Mullainathan, 2023). Indeed, multiple lines of work—spanning decades of research—have sought to extract text features that can predict a target variable (Blei et al., 2003; Monroe et al., 2009). Recent LLM-based hypothesis generation methods are especially exciting since they operate directly at the level of human-interpretable natural language concepts (Zhou et al., 2024; Batista & Ross, 2024). Concretely, hypothesis generation methods take a dataset of texts linked to a target variable (e.g., headlines and their engagement level) and hypothesize natural language concepts that predict the target variable (e.g., *mentions being surprised or shocked* or *asks a question to the reader*).

However, basic hurdles impede the full usage of LLMs for hypothesis generation. Consider two natural approaches. The first prompts an LLM with positive and negative examples and asks it to hypothesize concepts that distinguish them (Zhou et al., 2024; Batista & Ross, 2024; Ludan et al., 2024). However, due to context window and reasoning limitations, we can only give LLMs a few training examples at a time, making this hard to scale. A second approach aims to interpret a model fine-tuned to predict the target variable (Zhong et al., 2024), leveraging the full training dataset. However, neurons are often hard to interpret (Elhage et al., 2022). In summary, efficient and scalable learning of statistical relationships requires numerical representations of text (e.g., *neurons*), but interpreting neurons is hard.

Our first contribution is a theoretical framework that connects model interpretation and hypothesis generation, clarifying the challenge above. Proposition 3.1 gives a "triangle inequality" for hypothesis generation: if a neuron is predictive of the target variable, then a sufficiently high-fidelity interpretation of the neuron is also predictive. This result directly motivates a general hypothesis generation procedure:

1. (Feature generation) Learn interpretable neurons (i.e., that tend to fire in the presence of human concepts).
2. (Feature selection) Select neurons that predict the target variable.
3. (Feature interpretation) Generate high-fidelity natural language interpretations of these neurons.

[*] Equal contribution [1] UC Berkeley [2] Cornell University [3] Cornell Tech. Correspondence to: Rajiv Movva <rmovva@berkeley.edu>, Kenny Peng <kennypeng@cs.cornell.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

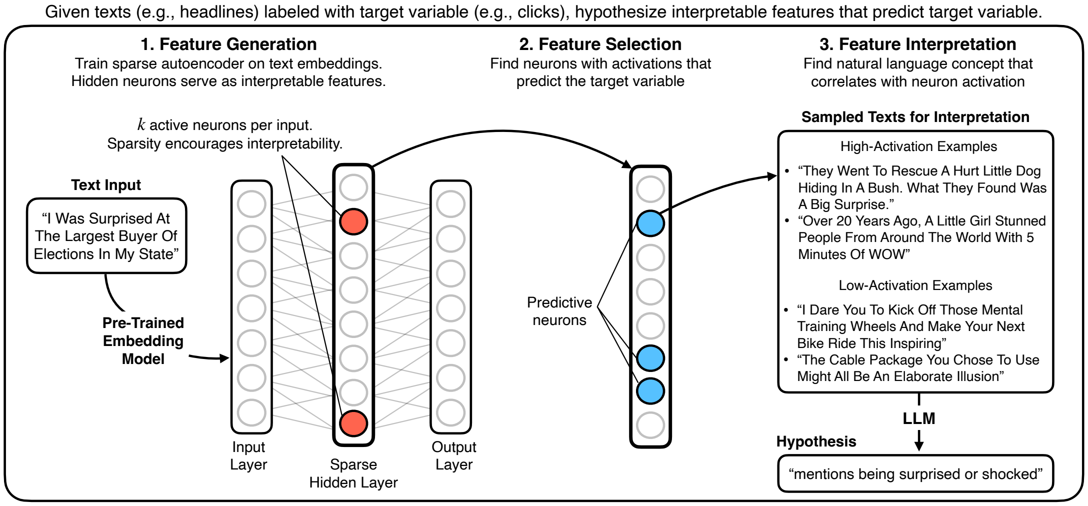
Given texts (e.g., headlines) labeled with target variable (e.g., clicks), hypothesize interpretable features that predict target variable.
1. Feature Generation
Train sparse autoencoder on text embeddings. Hidden neurons serve as interpretable features.
2. Feature Selection
Find neurons with activations that predict the target variable
3. Feature Interpretation
Find natural language concept that correlates with neuron activation
k active neurons per input. Sparsity encourages interpretability.
Text Input
"I Was Surprised At The Largest Buyer Of Elections In My State"
Pre-Trained Embedding Model
Input Layer
Sparse Hidden Layer
Output Layer
Predictive neurons
Sampled Texts for Interpretation
High-Activation Examples
• "They Went To Rescue A Hurt Little Dog Hiding In A Bush. What They Found Was A Big Surprise."
• "Over 20 Years Ago, A Little Girl Stunned People From Around The World With 5 Minutes Of WOW"
Low-Activation Examples
• "I Dare You To Kick Off Those Mental Training Wheels And Make Your Next Bike Ride This Inspiring"
• "The Cable Package You Chose To Use Might All Be An Elaborate Illusion"
LLM
Hypothesis
"mentions being surprised or shocked"

Figure 1. HypotheSAEs is a general method that outputs natural language hypotheses from text datasets.

Our second contribution is a method, HypotheSAEs, that successfully implements this procedure. A primary challenge is learning the interpretable neurons in step 1—after all, a long line of work has established the challenge of interpreting neural networks (Olah et al., 2017; Kim et al., 2018; Elhage et al., 2022). To overcome this, we train a sparse autoencoder (SAE; Makhzani & Frey (2014)) on pre-trained text embeddings from our dataset. Neurons in the SAE hidden layer are highly interpretable, while retaining the explanatory power of embeddings (Cunningham et al., 2023; Bricken et al., 2023; Templeton et al., 2024; Gao et al., 2024; O'Neill et al., 2024). Intuitively, this architecture leverages the fact that natural language is sparse; only a small fraction of all human concepts are expressed at a time. In step 2, we select neurons that are predictive of the target variable (e.g., using Lasso). In step 3, we automatically generate high-fidelity interpretations of these predictive neurons by prompting an LLM with texts that activate the neuron, building on recent work in autointerpretability (Bills et al., 2023). These interpretations serve as hypotheses.

We evaluate HypotheSAEs against recent state-of-the-art methods. We consider three synthetic tasks, as well as three real-world tasks of practical interest: hypothesizing the relationship between headlines and engagement, speech text and political party, and review text and rating. Example HypotheSAEs hypotheses are given below:

Task	Example Hypotheses (Abbreviated)
Headline Engagement	(+) *mentions being surprised or shocked* (-) *asks a question directly to the reader*
Speaker Party	(Rep.) *discusses illegal immigration* (Dem.) *criticizes tax breaks for the wealthy*
Restaurant Rating	(+) *mentions plans to return to the restaurant* (-) *mentions issues related to food safety*

HypotheSAEs produces many more significant hypotheses than three baseline methods. On three real-world tasks, 45/60 hypotheses generated by our method are significant, compared to at most 24 for the baselines. On two datasets, we identify new hypotheses which previous detailed analyses have not uncovered. Intuitively, by decoupling feature generation from the prediction task, HypotheSAEs is able to explore a wider range of hypotheses.

HypotheSAEs is also more than 10× faster and cheaper than recent LLM baselines, with similar runtime and cost to BERTopic, a standard embedding-based topic model. Filtering hypotheses based on the predictiveness of neurons is significantly cheaper than directly evaluating natural language concepts, since all neuron activations of an input can be computed from a single forward pass of an SAE (in comparison to one LLM call per natural language feature).

2. Related Work

2.1. Hypothesis Generation

Classical text analysis methods, such as comparing n-gram frequencies between groups (Fightin' Words; Monroe et al. (2009)) or topic modeling (LDA, Blei et al. (2003); BERTopic, Grootendorst (2022)) remain standard tools to discover relationships between text and target variables (Grimmer et al., 2022). A challenge with these methods is that their outputs—lists of words or documents—are not immediately interpretable by humans (Chang et al., 2009). As a result, applying these methods for scientific discovery requires sufficient domain expertise to prespecify hypotheses and assess whether the data support them (Demszky et al., 2019; Gentzkow et al., 2016; Sun et al., 2022).

Recently, several LLM-based methods aim to automate interpretable concept discovery from text datasets. Grootendorst (2024) prompts LLMs to generate cluster labels, Pham et al. (2024) use LLMs for end-to-end topic modeling in natural language, and Lam et al. (2024) build an LLM-driven interface to discover topics guided by user input. Another literature uses LLMs to propose concepts in bottleneck models (Koh et al., 2020): Ludan et al. (2024) prompt LLMs with a task description to generate predictive concepts, and Feng et al. (2024) use LLMs to provide a prior distribution over concepts. Sun et al. (2024) and Zhong et al. (2024) also use LLMs to propose concepts, guided by neurons from a supervised model. Finally, Zhou et al. (2024) and Batista & Ross (2024) focus specifically on hypothesis generation: LLMs propose natural language descriptions of predictive relationships, which can then be tested for generalization.

Unlike several of these methods, HypotheSAEs decouples feature generation (unsupervised, using the SAE), selection (supervised), and interpretation (the only LLM-dependent step). Comparing to baselines from each class of methods (§5.3), we illustrate quantitative and qualitative improvements over classical and fully-LLM-driven baselines.

2.2. Sparse Autoencoders

A recent line of work demonstrates that sparse autoencoders (Makhzani & Frey, 2014) are an effective architecture to learn interpretable features from intermediate layers of large language models (Cunningham et al., 2023; Bricken et al., 2023; Templeton et al., 2024; Bills et al., 2023). This literature is primarily motivated by mechanistic interpretability: since the neurons in the hidden layer of sparse autoencoders correspond to interpretable features, they can be strategically altered to steer the behavior of language models. In the present work, we instead apply sparse autoencoders to learn interpretable text features for the purpose of hypothesizing interpretable relationships between text data and a target variable. We follow O'Neill et al. (2024), who show that SAE features can guide semantic search applications, in training a sparse autoencoder directly on text embeddings.

3. Theoretical Framework

In this section, we derive a theoretical result that shows how a language model with highly interpretable neurons can be used for hypothesis generation, as well as where performance loss arises in such a procedure.

Let Z be an indicator for whether a neuron fires for a given text input (i.e., whether the activation exceeds a threshold). Let $\hat{Z}$ be an indicator for whether the text contains a specified natural language concept. Let Y be the target variable. Define the predictiveness of $\hat{Z}$ as the separation score

$$S(\hat{Z}) := |\mathbb{E}[Y|\hat{Z}=1] - \mathbb{E}[Y|\hat{Z}=0]|.$$

Define $S(Z)$ analogously.[1] We measure the fidelity of the interpretation $\hat{Z}$ by

$$\delta(\hat{Z}, Z) := \frac{1 - \min\{\text{recall}, \text{precision}\}}{\min\{\Pr[\hat{Z}=0], \Pr[Z=0]\}}, \qquad (1)$$

where precision is how often the neuron fires when the concept is present and recall is how often the concept is present when the neuron fires. Lower δ implies higher fidelity. High recall and precision means the interpretation approximates the neuron's behavior, and is not overly narrow or generic. Empirically, the denominator $\min\{\Pr[\hat{Z}=0], \Pr[Z=0]\}$ is close to 1 (neurons and concepts activate infrequently), so maximizing fidelity is approximately maximizing the minimum of recall and precision.

Proposition 3.1 (A Triangle Inequality for Hypothesis Generation)**.** *Suppose Y is supported on $[0, 1]$. Then*

$$|S(\hat{Z}) - S(Z)| \le \delta(\hat{Z}, Z). \qquad (2)$$

Intuitively, the result shows that if $\hat{Z}$ is a high-fidelity predictor of Z, then it will also have a similar separation score. A full proof is given in Appendix F.[2]

Notice that what we ultimately care about is $S(\hat{Z})$ (how predictive our actual natural language hypothesis is), but our procedure focuses primarily on generating, selecting, and then interpreting the neuron Z. Proposition 3.1 shows that as long as we can generate a sufficiently high-fidelity interpretation $\hat{Z}$ of each neuron Z, then we can focus on finding predictive neurons as a way to ultimately find natural language hypotheses. In particular, the result shows that

$$\begin{aligned} S(\hat{Z}) &= S(Z) + (S(\hat{Z}) - S(Z)) & (3)\\ &\ge \underbrace{S(Z)}_{\substack{\text{neuron}\\ \text{predictiveness}}} - \underbrace{\delta(\hat{Z}, Z)}_{\substack{\text{interpretation}\\ \text{fidelity}}}. & (4) \end{aligned}$$

Equation (4) shows that to find a natural language concept such that $S(\hat{Z})$ is large, it suffices to first identify Z such that $S(Z)$ is high (i.e., find a predictive neuron) and then $\hat{Z}$ such that $\delta(\hat{Z}, Z)$ is small (i.e., find a high-fidelity interpretation).

This directly motivates the actual procedure employed in HypotheSAEs. First, we learn interpretable neurons Z using an SAE. Second, we identify which neurons Z are

[1] Separation score is one way we evaluate hypotheses empirically. We suspect that analogs of Proposition 3.1 hold for alternative measures of predictiveness, like explained variance.

[2] There are two main observations that enable us to show the result. First, we focus on $|\mathbb{E}[Y|\hat{Z}=1] - \mathbb{E}[Y|Z=1]|$, and show that if the false negative rate (FNR) is greater than the false discovery rate (FDR), this value is bounded by the FNR; otherwise, it is bounded by the FDR. Second, we show roughly that as long as $\Pr[\hat{Z}=1], \Pr[Z=1] < \Pr[\hat{Z}=0], \Pr[Z=0]$, then $|\mathbb{E}[Y|\hat{Z}=0] - \mathbb{E}[Y|Z=0]| < |\mathbb{E}[Y|\hat{Z}=1] - \mathbb{E}[Y|Z=1]|$.

predictive. Third, we generate high-fidelity interpretations $\hat{Z}$ of these neurons with an LLM.

Equation (4) also gives us a lens by which to analyze HYPOTHESAEs empirically. The result decomposes the performance of hypotheses $\hat{Z}$ into two parts: the predictiveness of the neurons Z and the fidelity of the interpretations $\hat{Z}$. For example, if the SAE neurons are similarly predictive to the text embeddings they encode, then most performance loss (compared to predicting directly from embeddings) comes from low-fidelity neuron interpretation. Imperfect neuron interpretations can be due to either neurons being fundamentally uninterpretable (there doesn't *exist* a high-fidelity interpretation), or due to difficulties in actually *finding* the interpretation. These challenges arise in steps 1 (training the SAE) and step 3 (generating the interpretation) respectively. We perform such an empirical analysis in Appendix C.3.

4. Methods

In the previous section, we established theoretically that we can generate hypotheses by learning interpretable neurons, identifying which neurons are predictive, and then generating high-fidelity interpretations of the neurons. We now describe our method, HYPOTHESAEs, which accomplishes these steps in practice.

Given a dataset of training examples $\{(x_i, y_i)\}_{i\in[N]}$, where x_i is the input text and y_i is the target variable annotation, HYPOTHESAEs outputs H natural language concepts that serve as hypotheses. The goal is for these natural language concepts to predict the target variable.

4.1. Feature Generation: Training Sparse Autoencoders.

Let e_i denote a D-dimensional text embedding of x_i. We train a k-sparse autoencoder (Makhzani & Frey, 2014). k-sparse autoencoders have successfully generated interpretable features both when applied to intermediate layers of language models, as well as text embeddings (Gao et al., 2024; O'Neill et al., 2024). The SAE encodes a text embedding e_i as follows (we use OpenAI's text-embedding-3-small, where $D = 1536$, in our experiments):

$$z_i = \mathrm{ReLU}(\mathrm{TopK}(W_{\text{enc}}(e_i - b_{\text{pre}}) + b_{\text{enc}})), \quad (5)$$

$$\hat{e_i} = W_{\text{dec}} z_i + b_{\text{dec}}, \quad (6)$$

where $b_{\text{pre}} \in \mathbb{R}^D, W_{\text{enc}} \in \mathbb{R}^{M\times D}, b_{\text{enc}} \in \mathbb{R}^M, W_{\text{dec}} \in \mathbb{R}^{D\times M}, b_{\text{dec}} \in \mathbb{R}^D$, and TopK sets all activations except the top k to zero. In a basic k-sparse autoencoder, the loss on one input is

$$\mathcal{L} = ||e_i - \hat{e_i}||_2^2. \quad (7)$$

We follow Gao et al. (2024); O'Neill et al. (2024) in further adding an auxillary loss to avoid "dead latents." We describe details and hyperparameters in Appendix B.

Empirically, the dimensions of z_i are often highly interpretable, corresponding to human concepts. M sets the total number of concepts learned across the entire dataset, and k sets the number of concepts that can be used to reconstruct each instance. The output of this first step is an $N \times M$ activation matrix, Z_{SAE}.

4.2. Feature Selection: Target Prediction with SAE Neurons.

To select a predictive subset of SAE neurons, we fit an L_1-regularized linear or logistic regression predicting the target variable Y from the activation matrix Z_{SAE} (Tibshirani, 1996). Formally, for regression[3], we optimize

$$\min_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta}; \lambda) = \frac{1}{N} ||\mathbf{y} - Z_{\text{SAE}}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_1,$$

where $\mathbf{y}$ is a length-N vector, Z_{SAE} is an $N \times M$ matrix, and $\boldsymbol{\beta}$ is a length-M vector of feature coefficients for each SAE neuron. The L_1 penalty produces sparse coefficient vectors $\boldsymbol{\beta}$ where some coefficients are exactly zero (indicating a dropped feature). To generate H hypotheses, we perform binary search to identify a value of λ which produces exactly H nonzero coefficients.

4.3. Feature Interpretation: Labeling Neurons with LLMs.

In this step, we generate high-fidelity interpretations (i.e., natural language concepts) of the subset of predictive neurons. To generate interpretations of a neuron, we prompt an LLM with example texts from a range of neuron activations and instruct it to identify a natural language concept that is present in high-activation texts and absent in low-activation texts. We generate multiple interpretations by running this procedure several times, and choose the best interpretation according to a measure of fidelity we describe below.

We test a variety of approaches for generating interpretations inspired by recent literature on automated interpretability (Bills et al., 2023; Templeton et al., 2024). In our final procedure, we sample high- and low-activating texts from the neuron's positive activation distribution, with the sampling percentile bins given in Appendix B.2. (Compared to zero-activating texts, texts with a low but nonzero activation act as hard negatives, improving the generated concept's precision.) Treating these as true positives and true negatives, respectively, we prompt an LLM to generate a natural language concept that distinguishes 10 samples of each class. We then define an interpretation's fidelity as its F1 score in this prediction task, using an LLM to annotate 100 positive and 100 negative samples for the presence of the concept.

[3] For classification we analogously use an L_1-regularized loss: $\mathcal{L}(\boldsymbol{\beta}; \lambda) = \frac{1}{N}\,\mathrm{BCE}(\mathbf{y}, \sigma(Z_{\text{SAE}}\boldsymbol{\beta})) + \lambda||\boldsymbol{\beta}||_1$, where BCE is the binary cross-entropy loss and $\sigma(\cdot)$ is the sigmoid function.

We use GPT-4o with temperature 0.7 to generate concepts and GPT-4o-mini with temperature 0 for concept annotation.

This approach produced predictive interpretations when compared to other prompting and evaluation schemes, though some other methods performed similarly (see Appendix C for details). Importantly, we found experimentally that higher-fidelity interpretations according to our measure have better predictive performance (Appendix C.1), justifying our approach to selecting interpretations.

5. Experiments

5.1. Datasets

We evaluate on synthetic and real-world datasets, described below. Appendix A provides more details on all datasets.

Synthetic datasets: recovering known, reference hypotheses. Our synthetic evaluation is motivated by real-world settings in which there are multiple disjoint hypotheses we would like to discover. We therefore use two datasets from prior work on interpretable clustering (Pham et al., 2024; Zhong et al., 2024): **WIKI** and **BILLS**. We construct a target variable that is positive if a text belongs to one of a pre-specified list of reference categories; these are the ground-truth "hypotheses" to recover. Both datasets are labeled by humans with granular hierarchical topics, such as *Media and Drama: Television: The Simpsons* in WIKI and *Macroeconomics: Tax Code* in BILLS.

We generate ground-truth labels based on the most frequent granular topics: articles in any of the top-5 topics are labeled '1', and all others are labeled '0'. This produces 5 reference hypotheses, e.g., *the bill is about Macroeconomics: Tax Code* is one of them. For WIKI, we also include a more challenging variation of recovering the top-15 most frequent topics. WIKI contains 11,979 total articles (15% positive for WIKI-5; 35% for WIKI-15), and BILLS contains 21,149 (24% positive). For both datasets, we reserve 2,000 items for validation (i.e., SAE hyperparameter selection) and 2,000 heldout items to evaluate hypotheses; we use the remaining items for SAE training and feature selection.

Real-world datasets: Generating hypotheses that predict a target variable. We apply HYPOTHESAEs to three real-world datasets which have been studied in prior work:

- HEADLINES (Matias et al., 2021): Which features of digital news headlines predict user engagement? Each instance is a pair of differently-phrased headlines for the same article; users on Upworthy.com were randomly shown one, and the task is to identify which headline received higher click-rate. We reserve a large heldout set to improve statistical power: our split sizes are 8.8K training, 1K validation, and 4.4K heldout.
- YELP (Yelp, 2024): Which features of Yelp restaurant reviews predict users' 1-5 star ratings? We use 200K reviews for training, 10K for validation, and 10K for heldout eval.
- CONGRESS (Gentzkow & Shapiro, 2010): Which features of U.S. congressional speeches predict party affiliation? Speeches are from the 109th Congress (2005–07) with binary labels (Rep. or Dem.). Our split sizes are 114K training, 16K validation, and 12K heldout.

On each of these tasks, our goal is to identify interpretable natural language features that predict the target variable. The former two are studied in prior work on hypothesis generation (Zhou et al., 2024; Batista & Ross, 2024; Ludan et al., 2024); the latter is a longstanding application in computational social science (Grimmer et al., 2021).

5.2. Evaluation Metrics

Synthetic datasets. On the synthetic datasets, we evaluate how well we recover the reference hypotheses. We run HYPOTHESAEs by setting H to the number of references (e.g., 5 for WIKI-5). We use GPT-4o-mini to annotate each inferred hypothesis on the heldout set, producing an $N_{\text{heldout}} \times H$ binary matrix of annotations. Following Zhong et al. (2024), we compute the optimal matching between reference and inferred hypotheses via the Hungarian algorithm on the $H \times H$ correlation matrix of reference vs. inferred hypothesis annotations, and we report three metrics:

- **F1 Similarity**: For each matched (reference, inferred) pair, we compute the F1-score between the presence of the ground-truth reference topic and the inferred hypothesis annotations. We report the mean across the H pairs.
- **Surface Similarity**: For each matched pair, we prompt GPT-4o to assess whether the hypotheses are the same, related, or distinct, with corresponding scores of 1.0, 0.5, and 0.0. To improve stability, we sample 5 outputs at temperature 0.7 and average them. We report the mean value of this score across the H pairs. (Table 4 shows examples; Figure 6 provides the prompt.)
- **Overall AUC**: Using annotations from the H inferred hypotheses, we fit a multivariate logit and compute AUC to evaluate predictive performance. (Recall that the ground-truth label is a logical OR of the reference topics.)

The first and second metrics are taken directly from Zhong et al. (2024); they assess whether the inferred hypotheses quantitatively and qualitatively match the ground-truth. The third metric assesses overall prediction quality. Some hypotheses may not match a reference but still be predictive,

Source	Overall AUC	# Sig	Significant Hypotheses: Green: ↑ clicks ; Red: ↓ clicks	Sep.	AUC
HypotheSAEs	0.693	15	*mentions photography, images, or visual representations*	0.26	0.51
			mentions situations or behaviors that evoke discomfort or embarrassment	0.16	0.58
			includes a reference to being surprised, shocked, or unprepared	0.16	0.56
			focuses on a female subject and her personal journey	0.14	0.52
			mentions or heavily emphasizes a video or video-related content	0.13	0.53
			mentions a surprising or unexpected action or outcome	0.13	0.58
			focuses on a male protagonist or subject	0.11	0.53
			contains phrases about looking or seeing something	0.09	0.52
			mentions uniqueness or exclusivity using superlative or comparative language	0.09	0.53
			mentions a response to or confrontation with offensive behavior	0.09	0.53
			mentions politicians, government actions, or political topics	-0.10	0.52
			asks a question directly to the reader	-0.12	0.53
			addresses collective human responsibility or action	-0.13	0.56
			mentions environmental issues or ecological consequences	-0.15	0.51
			mentions actions or initiatives that positively impact a community	-0.16	0.53
BERTopic	0.619	1	*headlines emphasize emotional reactions to short, impactful videos*	0.15	0.59
NLParam	0.660	5	*has an element of surprise in its structure*	0.14	0.57
			frames the content within a gender-specific context	0.11	0.53
			includes an inspiring transformation	-0.08	0.51
			uses a play on words or puns	-0.09	0.54
			poses a rhetorical question	-0.12	0.55
HypoGeniC	0.646	3	*implies a personal story or anecdote that invites curiosity*	0.15	0.56
			uses strong emotional language or evokes curiosity	0.12	0.58
			poses a rhetorical question that may disengage the reader	-0.11	0.55

Table 1. Hypotheses on Headlines with $p < 0.05$ after Bonferroni correction in a multivariate regression. 'Sep.' is the separation score (§3): $E[Y|\hat{Z}=1] - E[Y|\hat{Z}=0]$, where positive scores predict higher click-rates. The 'AUC' column provides AUCs of individual hypotheses. 'Overall AUC' uses all 20 hypotheses, including insignificant ones. Hypotheses for all methods are abbreviated for space.

so this metric rewards any interpretable hypotheses that contribute predictive value, even if they are not optimal.

Real datasets. On the real datasets, we generate 20 hypotheses per method, and assess the hypothesis sets for two qualities: (1) *breadth*: how many distinct, predictive hypotheses were identified? (2) *predictiveness*: how well do the hypotheses collectively explain the label? We again compute an $N_{\text{heldout}} \times H$ annotation matrix for each method's hypotheses, and measure these constructs as follows:

- **Breadth**: We fit a multivariate logit (for binary labels) or OLS (continuous) regressing the target variable against all of the hypothesis annotations, and report the count of hypotheses that have a significant, nonzero coefficient[4]. This metric benefits distinct hypotheses and penalizes redundancy. Breadth is important since the utility of hypotheses is often determined by criteria beyond predictiveness. For example, a hypothesis may be more useful if it is connected to past theories or suggests a new direction for investigation. More breadth increases the likelihood that a hypothesis satisfying these criteria is produced.

- **Overall Predictive Performance**: Similar to the synthetic datasets, we report the overall prediction quality using all of the hypotheses in a shared regression, reporting AUC for classification and R^2 for regression.

- **Qualitative Annotation**: We recruit three computational science researchers to rate each hypothesis for (1) helpfulness and (2) interpretability (details in App. E). These attributes are inspired by prior work (Lam et al., 2024).

5.3. Baselines

We compare HypotheSAEs against recent, state-of-the-art methods (§2). For fair comparison, we run all methods with the same embeddings and LLMs as our method, unless stated otherwise: OpenAI's text-embedding-3-small for embeddings; GPT-4o for hypothesis proposal; and GPT-4o-mini for hypothesis annotation. We run each method to produce H hypotheses and evaluate all methods identically.

BERTopic (Grootendorst, 2022) is a neural topic model that produces topics by clustering text embeddings. To produce H hypotheses from BERTopic, texts are clustered into topics, and we fit an L_1-regularized model to predict Y from topic ID; like our method, we set the penalty such that there are exactly H nonzero coefficients. We assign each topic a natural language label by prompting an LLM with a sample of documents and top terms (prompt in Figure 8).

NLParam (Zhong et al., 2024) trains a neural network

[4] We use a Bonferroni-corrected p-value threshold of 2.5e-3.

	Wiki, $H = 5$			Wiki, $H = 15$			Bills, $H = 5$			Headlines		Yelp		Congress	
Method	F1	Surf.	AUC	F1	Surf.	AUC	F1	Surf.	AUC	Ct.	AUC	Ct.	R^2	Ct.	AUC
HypotheSAEs	**0.58**	**0.70**	**0.84**	**0.47**	**0.61**	**0.84**	**0.67**	0.78	**0.89**	**15**	**0.69**	**15**	0.76	**15**	**0.70**
BERTopic	0.48	0.48	0.77	0.40	0.43	0.79	0.65	**0.88**	0.85	1	0.62	12	0.65	10	0.64
NLParam	0.15	0.30	0.69	0.16	0.34	0.77	0.32	0.50	0.74	5	0.66	11	0.58	8	0.65
HypoGeniC	0.15	0.20	0.60	0.12	0.39	0.64	0.25	0.30	0.70	3	0.65	12	**0.78**	5	0.68

Table 2. HypotheSAEs performs best at recovering reference human-annotated topics on Wiki-5, Wiki-15, and Bills. We also evaluate on three real-world datasets: Headlines, Yelp, and Congress. "Ct." refers to the count of significant hypotheses, out of 20 candidates; HypotheSAEs generates the most significant hypotheses.

with a length-K bottleneck layer to predict Y from text embeddings, and then uses an LLM to propose labels corresponding to each one of the bottleneck neurons' activations. The hypotheses are iteratively refined; a hypothesis is retained only if its annotations (computed on the entire dataset) improve the loss. We run NLParam for the default 10 iterations, with 3 candidate hypotheses per neuron.

HypoGeniC (Zhou et al., 2024) directly prompts a language model to propose hypotheses given a task-specific prompt and labeled examples. They use a bandit reward function to score hypotheses and propose new ones based on incorrectly-labeled examples. We run HypoGeniC with default parameters; note that the method uses the same model for proposal and annotation, which defaults to GPT-4o-mini.

Training set size. A key advantage of HypotheSAEs is that feature generation and selection require only an SAE forward pass, which is cheap even for large datasets; the only LLM-dependent step is feature interpretation. BERTopic also scales well, as it only uses LLM inference for topic labeling. In contrast, NLParam and HypoGeniC score and select hypotheses by annotating every training example, so their inference costs scale with training set size. Running either method on 200K Yelp reviews, for instance, would cost >\$500 and 78 hours (compared to ∼\$1 and 20 minutes for HypotheSAEs). To compare to the baselines as favorably as possible, we allow each method to use up to 50× the cost and runtime of ours. To fit this budget, we use up to 20,000 training samples for NLParam and HypoGeniC, which accommodates the full training sets on Wiki, Bills, and Headlines; we use random subsets for Yelp and Congress. Notably, this sample budget is significantly larger than either method uses in their publication experiments (4,748 for NLParam; 200 for HypoGeniC), suggesting our comparisons are generous to the baselines.

6. Results

6.1. Synthetic Datasets

HypotheSAEs recovers ground-truth hypotheses. Table 2 shows quantitative metrics: HypotheSAEs beats all baselines on 8 of 9 dataset-metric pairs. As captured by surface similarity, most of HypotheSAEs's inferred hypotheses either perfectly match or are related to the references. On Wiki-5 (Table 4), we retrieve two hypotheses exactly; the other three are slightly too specific (HypotheSAEs infers a topic *is about a New York State Route or highway*, but the reference includes all Northeastern roads, not just New York) or broad (it infers *historical battles* but does not specify *battles since 1800*), but all inferred hypotheses are always related to a reference, i.e., has surface similarity ≥ 0.5. No baselines meet this standard on Wiki-5: NLParam and HypoGeniC output 3/5 and 2/5 hypotheses that are unrelated to any of the reference topics, respectively. BERTopic performs well, but outputs redundant hypotheses about state highways and misses 'battles' entirely.

Our hypotheses also closely match ground-truth when used to annotate heldout data. The mean F1 score compares how closely annotations according to an inferred hypothesis (e.g. *is about a Simpsons episode*) match ground-truth labels (whether the Wiki article belongs to the human-annotated Simpsons category). HypotheSAEs beats all baselines on this metric, especially outperforming NLParam (+0.36, on average) and HypoGeniC (+0.40). Finally, regressing the label using each method's annotations, HypotheSAEs achieves higher AUC than all baselines. Overall, HypotheSAEs usually outperforms BERTopic—which is designed exactly for this topic inference setting—and substantially outperforms two state-of-the-art LLM hypothesis generation methods.

6.2. Real-World Datasets

Table 2 shows quantitative metrics for the real-world datasets. Tables 1, 5, and 6 show all significant hypotheses on Headlines, Congress, and Yelp respectively.

HypotheSAEs identifies more distinct hypotheses than other methods. Compared to baselines, HypotheSAEs produces the most hypotheses which are significant in a multivariate regression. Importantly, this evaluation reflects successful *generalization*: the hypotheses are learned on the training and validation sets, and they remain significantly associated with the target variable when annotated by GPT-4o-mini on the heldout set. Across the three datasets, **45 out of 60 candidate hypotheses are significant**, substantially ahead of the 24, 23, and 20 for NLPARAM, BERTOPIC, and HYPOGENIC respectively.

When evaluating overall predictive performance, HYPOTHESAES beats baselines on 8 of 9 comparisons, demonstrating strongly predictive hypotheses. The exception is that HYPOGENIC achieves 2% higher R^2 on Yelp. However, HYPOGENIC's most predictive hypotheses, such as *expresses disappointment with the overall dining experience*, essentially restate the rating prediction task. While most LLMs can perform such a prediction, these hypotheses do not reveal specific insights about the task. In contrast, our hypotheses—in Table 6 for YELP—tend to be more specific: we find that negative reviews mention food poisoning, rude and aggressive staff, or dishonest business practices.

HYPOTHESAES's outputs are rated as qualitatively helpful and interpretable. The results of our qualitative evaluation (Figure 2) confirm the above intuition: most of the predictive concepts generated by HYPOTHESAES are rated helpful (24 out of 30) and interpretable (29 out of 30)[5]. Several hypotheses from NLPARAM and HYPOGENIC, despite being predictive, do not pass these checks. The total count of hypotheses that are predictive, helpful, and interpretable generated by HYPOTHESAES (23) far exceeds any baseline (BERTOPIC is second-best with 7).

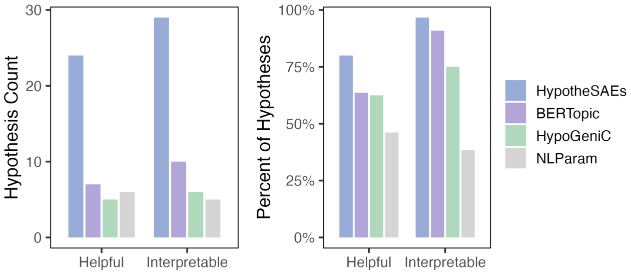

Figure 2. HYPOTHESAES produces a larger *number* (left) and *percentage* (right) of significant hypotheses that are rated helpful and interpretable. We report majority vote of 3 ratings per hypothesis.

6.3. Non-SAE Ablations

To isolate the value of the SAE, we conduct ablation experiments. Because our method differs from baselines in multiple ways—e.g., our neuron interpretation procedure differs from that of NLPARAM—it is worth asking what gain the SAE specifically provides. To answer this question, we run the following ablations on the real-world datasets:

- **Embedding**: We skip training an SAE and directly select 20 features, via Lasso, from text embeddings. The rest of our method is unchanged.
- **PCA**: Same as **Embedding**, but we first use PCA to reduce the embedding dimension from 1536 down to 256.
- **Bottleneck**: Inspired by bottleneck models (Yang et al., 2023), we fit a neural network to predict y using text embeddings as input and a single hidden layer with 20 neurons. We generate hypotheses by interpreting the hidden neurons (in the same way we interpret SAE neurons).

We show results in Table 7. In addition to the count of significant hypotheses and overall predictiveness, we also report average neuron fidelity.

We establish two benefits of the SAE: the neuron interpretations are (1) higher fidelity and (2) cover a wider variety of non-redundant concepts. Regarding (1), interpretations of SAE neurons achieve a mean F1 of 0.84, compared to 0.64 for the best ablation (bottleneck). Regarding (2), we see improvements to downstream metrics: the SAE produces many more significant hypotheses (45/60 across the three datasets, vs. 30/60 for the best ablation) and achieves slightly higher predictiveness. Qualitatively, the SAE neurons yield more specific interpretations, while the ablations tend to yield generic ones. Both benefits (1) and (2) are consistent with the SAE producing monosemantic neurons, which are easy to interpret and specific enough to avoid redundancy.

6.4. Evaluating Hypothesis Novelty

Beyond quantitative comparisons against baselines, we evaluate if HYPOTHESAES discovers new hypotheses even on datasets that have been thoroughly analyzed in prior work.

CONGRESS. A classic study from Gentzkow et al. (2016) identifies bigrams and trigrams in Congressional speeches that mark Republican (R) or Democrat (D) speakers. We compare to their work by computing counts, in each speech, of their full reported n-gram lists (150 R / 150 D); we also run HYPOTHESAES to produce 100 hypotheses, of which 33 are significant. A regression using only n-gram counts achieves AUC 0.61, while including our 33 hypotheses increases AUC to 0.74. Notably, 28 out of 33 hypotheses remain significantly predictive when controlling for n-grams, showing that our hypotheses add signal to classical methods.

Beyond predictive performance, HYPOTHESAES offers two qualitative advantages. First, our hypotheses are inherently interpretable. For example, while Gentzkow et al. (2016) report that "oil companies" correlates with Democrats, it requires context to interpret: are speakers referring to oil companies positively or negatively? In contrast, HYPOTHESAES's corresponding hypothesis is *criticizes oil companies or oil industry practices*. Second, our hypotheses capture nuanced patterns that n-grams cannot expresss, like *criticizes government resource allocation, highlighting disparities between domestic needs and actions abroad*.

[5] To reduce annotator workload, our qualitative evaluation only included results from HEADLINES and CONGRESS, since these are more well-studied social science tasks than YELP.

Headlines. To test our discoveries against more modern methods, we compare to Batista & Ross (2024), an LLM hypothesis generation study designed for Headlines and aimed at producing new insights for marketing researchers. Each of their 5 hypotheses is validated via human annotation. Using our heldout evaluation protocol, we compute annotations using their hypotheses, which achieve AUC 0.59 at predicting binary engagement; adding in HypotheSAEs's 15 significant hypotheses increases AUC to 0.70. Crucially, **all 15 of our hypotheses remain significant** when controlling for the prior discoveries in a shared regression.

Several other prior works have also studied Headlines using classical dictionary-based methods (Robertson et al., 2023; Gligorić et al., 2023; Banerjee & Urminsky, 2024; Aubin Le Quéré & Matias, 2025). HypotheSAEs qualitatively improves on these findings by producing more specific hypotheses. For example, Robertson et al. (2023) find that negative words increase engagement, and positive words decrease it. HypotheSAEs supports this finding, and also make it more precise: we find that *situations that evoke discomfort or embarrassment* and mentions of offensive behavior increase clicks, while *initiatives that positively impact a community* decreases them (Table 1). Banerjee & Urminsky (2024) find that words in an “emotional intensity” dictionary increase engagement, which may be difficult to operationalize; our corresponding finding is that headlines referencing *being surprised, shocked, or unprepared* increase engagement. Our findings are more precise, and can therefore provide richer insights to researchers.

These results demonstrate that HypotheSAEs, despite no dataset-specific design, uncovers novel hypotheses that advance prior work from domain experts—suggesting broad applicability across diverse settings.

6.5. Costs

In Table 3, we report the runtimes, LLM inference token counts, and costs (at current OpenAI API pricing) for all methods on Congress. The trends are similar for all datasets. Runtimes include training the SAEs on one NVIDIA A6000 GPU. Training is fast because our SAEs can afford to be small ($\sim 10^3$ features, compared to 10^7 in Gao et al. (2024)), since our focus is to learn domain-specific features rather than features of all language.

HypotheSAEs is much more efficient than NLParam and HypoGeniC. Despite the fact that these methods are trained on less data—20K examples vs. 114K for HypotheSAEs—**HypotheSAEs is $\sim$30-50× faster and 10-50× cheaper**. This is because NLParam and HypoGeniC score hypotheses by annotating every example in the dataset. As a result, their LLM annotation requirements scale with both the number of hypotheses H and dataset size N, while HypotheSAEs's scales only with H.

Method	Time (min)	Tokens (M)	Cost
HypotheSAEs	18.1	0.2 / 7.2	$1.29
Sae-NoVal	7.9	0.1 / 0.0	$0.15
NLParam	904.9	2.8 / 414	$69.92
HypoGeniC	575.4	0.0 / 72	$11.02
BERTopic	3.0	0.1 / 0.0	$0.19

Table 3. Runtimes and costs for hypothesis generation on Congress. We report millions of tokens for GPT-4o (concept generation) and GPT-4o-mini (concept annotation) respectively as {# 4o} / {# 4o-mini}. Input and output token counts are summed (>99% are input tokens). Sae-NoVal is our method without the fidelity validation step (i.e., for each neuron, we generate only one label instead of choosing from three based on fidelity). Costs are in USD and based on OpenAI pricing as of 01/2025.

BERTopic is cheapest, and remains a good baseline for a resource-limited analysis. However, HypotheSAEs can also be made cheaper by skipping the non-essential label validation step: instead of generating 3 candidate labels per neuron and choosing the highest-fidelity one, we can simply generate 1 label per neuron and use it directly. This reduces cost to that of BERTopic, while still exceeding its performance; we describe this ablation in Appendix B.2.

7. Conclusion

We propose HypotheSAEs, a scalable method to generate interpretable hypotheses from black-box text representations. On three synthetic and three real-world tasks, HypotheSAEs outperforms state-of-the-art topic modeling and LLM baselines, at less than 10× the time and cost of the latter. Beyond the tasks we study, the method applies naturally to the many settings in the social sciences where a large text corpus is labeled with a target variable of interest (Ziems et al., 2024; Bollen et al., 2011; Card et al., 2022; Breuer et al., 2025). In future work, an exciting direction is extending our method to explain non-text datasets, like images (Dunlap et al., 2024) and proteins (Vig et al., 2021), as well as text datasets from specialized domains like healthcare (Hsu et al., 2023; Robitschek et al., 2025). More generally, there are many applications where we can accurately *predict* a target variable, but producing new scientific insights remains difficult. Our work helps bridge this gap.

Software and Data

Code is available on GitHub, and can be installed via pip with `pip install hypothesaes`. A notebook to reproduce experimental results in the paper is available in the repository. All data used in the paper are available on HuggingFace. A demo to visualize all SAE neurons and how well they predict the target for Headlines, Yelp, and Congress is available at `hypothesaes.org`.

Acknowledgements

Thanks to Rishi Jha, Hal Triedman, Anna Lyubarskaja, Ruiqi Zhong, Serina Chang, and Sammi Cheung for helpful discussions. Thanks to Sidhika Balachandar, Divya Shanmugam, and Gabriel Agostini for qualitative annotation. Thanks to the ICML reviewers for helpful suggestions (specifically, to include SAE ablations and qualitative evaluation). Thanks to Neil Movva and OpenAI for LLM inference credits.

Funding: RM is supported by NSF DGE #2146752. NG is supported by NSF CAREER #2339427, and Cornell Tech Urban Tech Hub, Meta, and Amazon research awards. JK is supported by a Simons Collaboration grant and a grant from the MacArthur Foundation. EP is supported by a Google Research Scholar award, an AI2050 Early Career Fellowship, NSF CAREER #2142419, a CIFAR Azrieli Global scholarship, a gift to the LinkedIn-Cornell Bowers CIS Strategic Partnership, the Survival and Flourishing Fund, Open Philanthropy, and the Abby Joseph Cohen Faculty Fund.

Impact Statement

Our work advances the field of AI-assisted hypothesis generation, which we hope will drive progress in the social and natural sciences. To the best of our knowledge, there are no particular negative social consequences imposed by our work compared to machine learning research in general.

References

Aubin Le Quéré, M. and Matias, J. N. When curiosity gaps backfire: Effects of headline concreteness on information selection decisions. *Scientific Reports*, 15(1):994, January 2025. ISSN 2045-2322.

Banerjee, A. and Urminsky, O. The Language That Drives Engagement: A Systematic Large-scale Analysis of Headline Experiments. *Marketing Science*, November 2024. ISSN 0732-2399.

Batista, R. M. and Ross, J. Words that Work: Using Language to Generate Hypotheses, July 2024.

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

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022, March 2003. ISSN 1532-4435.

Bollen, J., Mao, H., and Zeng, X.-J. Twitter mood predicts the stock market. *Journal of Computational Science*, 2 (1):1–8, March 2011. ISSN 18777503.

Breuer, A., Dietrich, B. J., Crespin, M. H., Butler, M., Pyrse, J. A., and Imai, K. Using AI to Summarize US Presidential Campaign TV Advertisement Videos, 1952-2012, March 2025. Comment: 17 pages, 7 tables, 4 figures, and linked datasets.

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

Card, D., Chang, S., Becker, C., Mendelsohn, J., Voigt, R., Boustan, L., Abramitzky, R., and Jurafsky, D. 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, August 2022.

Chang, J., Gerrish, S., Wang, C., Boyd-graber, J., and Blei, D. Reading Tea Leaves: How Humans Interpret Topic Models. In *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Demszky, D., Garg, N., Voigt, R., Zou, J., Gentzkow, M., Shapiro, J., and Jurafsky, D. Analyzing Polarization in Social Media: Method and Application to Tweets on 21 Mass Shootings, April 2019. Comment: NAACL 2019; code and data available at https://github.com/ddemszky/framing-twitter.

Dunlap, L., Zhang, Y., Wang, X., Zhong, R., Darrell, T., Steinhardt, J., Gonzalez, J. E., and Yeung-Levy, S. Describing Differences in Image Sets with Natural Language, April 2024. Comment: CVPR 2024 Oral.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M., and Olah, C. Toy Models of Superposition, September 2022. Comment: Also available at https://transformer-circuits.pub/2022/toy_model/index.html.

Feng, J., Kothari, A., Zier, L., Singh, C., and Tan, Y. S. Bayesian Concept Bottleneck Models with LLM Priors, October 2024.

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

Gentzkow, M. and Shapiro, J. M. What Drives Media Slant? Evidence From U.S. Daily Newspapers. *Econometrica*, 78(1):35–71, 2010. ISSN 1468-0262.

Gentzkow, M., Shapiro, J. M., and Taddy, M. Measuring Group Differences in High-Dimensional Choices: Method and Application to Congressional Speech, July 2016.

Gligorić, K., Lifchits, G., West, R., and Anderson, A. Linguistic effects on news headline success: Evidence from thousands of online field experiments (Registered Report). *PLOS ONE*, 18(3):e0281682, March 2023. ISSN 1932-6203.

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

Grimmer, J., Roberts, M. E., and Stewart, B. M. Machine Learning for Social Science: An Agnostic Approach. *Annual Review of Political Science*, 24(Volume 24, 2021): 395–419, May 2021. ISSN 1094-2939, 1545-1577.

Grimmer, J., Roberts, M. E., and Stewart, B. M. *Text as Data: A New Framework for Machine Learning and the Social Sciences*. Princeton University Press, January 2022. ISBN 978-0-691-20799-5.

Grootendorst, M. BERTopic: Neural topic modeling with a class-based TF-IDF procedure, March 2022. Comment: BERTopic has a python implementation, see https://github.com/MaartenGr/BERTopic.

Grootendorst, M. LLM & Generative AI - BERTopic. https://maartengr.github.io/BERTopic/getting_started/representation/llm.html, 2024.

Hoyle, A. M., Sarkar, R., Goel, P., and Resnik, P. Are Neural Topic Models Broken? In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 5321–5344. Association for Computational Linguistics, December 2022.

Hsu, C.-C., Obermeyer, Z., and Tan, C. Clinical Notes Reveal Physician Fatigue, December 2023.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., and Sayres, R. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, pp. 2668–2677. PMLR, July 2018.

Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., and Liang, P. Concept Bottleneck Models, December 2020. Comment: Edited for clarity from the ICML 2020 version.

Lam, M. S., Teoh, J., Landay, J., Heer, J., and Bernstein, M. S. Concept induction: Analyzing unstructured text with high-level concepts using lloom, 2024. URL https://doi.org/10.1145/3613904.3642830.

Ludan, J. M., Lyu, Q., Yang, Y., Dugan, L., Yatskar, M., and Callison-Burch, C. Interpretable-by-Design Text Understanding with Iteratively Generated Concept Bottleneck, April 2024.

Ludwig, J. and Mullainathan, S. Machine Learning as a Tool for Hypothesis Generation, March 2023.

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

Matias, J. N., Munger, K., Le Quere, M. A., and Ebersole, C. The Upworthy Research Archive, a time series of 32,487 experiments in U.S. media. *Scientific Data*, 8(1): 195, August 2021. ISSN 2052-4463.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer Sentinel Mixture Models, September 2016.

Monroe, B. L., Colaresi, M. P., and Quinn, K. M. Fightin' Words: Lexical Feature Selection and Evaluation for Identifying the Content of Political Conflict. *Political Analysis*, 16(4):372–403, August 2009. ISSN 1047-1987, 1476-4989.

Nelson, L. K. Computational grounded theory: A methodological framework. *Sociological Methods & Research*, 49(1):3–42, 2020.

Olah, C., Mordvintsev, A., and Schubert, L. Feature Visualization. *Distill*, 2(11):10.23915/distill.00007, November 2017. ISSN 2476-0757.

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

Pham, C. M., Hoyle, A., Sun, S., Resnik, P., and Iyyer, M. TopicGPT: A Prompt-based Topic Modeling Framework, April 2024.

Ranard, B. L., Werner, R. M., Antanavicius, T., Schwartz, H. A., Smith, R. J., Meisel, Z. F., Asch, D. A., Ungar, L. H., and Merchant, R. M. Yelp reviews of hospital care can supplement and inform traditional surveys of the patient experience of care. *Health Affairs*, 35(4):697–705, 2016.

Rathje, S., Van Bavel, J. J., and Van Der Linden, S. Outgroup animosity drives engagement on social media. *Proceedings of the National Academy of Sciences*, 118(26): e2024292118, 2021.

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

Robitschek, E., Bastani, A., Horwath, K., Sordean, S., Pletcher, M. J., Lai, J. C., Galletta, S., Ash, E., Ge, J., and Chen, I. Y. A large language model-based approach to quantifying the effects of social determinants in liver transplant decisions, January 2025. Comment: Spotlight Paper, ML4H 2024; Leonidas H. Berry Health Equity Research Award, ACG 2024; Plenary Presentation, AASLD 2024.

Sun, C.-E., Oikarinen, T., Ustun, B., and Weng, T.-W. Concept Bottleneck Large Language Models, December 2024. Comment: arXiv admin note: substantial text overlap with arXiv:2407.04307.

Sun, M., Oliwa, T., Peek, M. E., and Tung, E. L. Negative Patient Descriptors: Documenting Racial Bias In The Electronic Health Record. *Health Affairs*, 41(2):203–211, February 2022. ISSN 0278-2715.

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

Tibshirani, R. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, January 1996. ISSN 0035-9246.

Ting, P.-J. L., Chen, S.-L., Chen, H., and Fang, W.-C. Using big data and text analytics to understand how customer experiences posted on yelp. com impact the hospitality industry. *Contemporary Management Research*, 13(2), 2017.

Vig, J., Madani, A., Varshney, L. R., Xiong, C., Socher, R., and Rajani, N. F. BERTology Meets Biology: Interpreting Attention in Protein Language Models, March 2021. Comment: To appear in ICLR 2021.

Yang, Y., Panagopoulou, A., Zhou, S., Jin, D., Callison-Burch, C., and Yatskar, M. Language in a Bottle: Language Model Guided Concept Bottlenecks for Interpretable Image Classification, April 2023. Comment: Published in CVPR 2023, 18 pages, 12 figures, 16 tables.

Yelp. Yelp Open Dataset. https://business.yelp.com/data/resources/open-dataset/, 2024.

Zhong, R., Wang, H., Klein, D., and Steinhardt, J. Explaining Datasets in Words: Statistical Models with Natural Language Parameters, September 2024.

Zhou, Y., Liu, H., Srivastava, T., Mei, H., and Tan, C. Hypothesis Generation with Large Language Models, August 2024.

Ziems, C., Held, W., Shaikh, O., Chen, J., Zhang, Z., and Yang, D. Can Large Language Models Transform Computational Social Science?, February 2024. Comment: To appear in ”Computational Linguistics” (CL).

Reference	Inferred	F1	Surface
Other music articles > Performers, groups, composers, and other music-related people	*is about a rock band or a musical act*	0.39	0.5
Earth science > Tropical cyclones: Atlantic	*is about hurricanes or tropical storms*	0.52	1.0
Battles, exercises, and conflicts > Modern history (1800 to present)	*is about a historical battle*	0.36	0.5
Television > The Simpsons episodes	*is about episodes of The Simpsons television show*	0.92	1.0
Transport > Road infrastructure: Northeastern United States	*is about a New York State Route*	0.71	0.5

Table 4. Evaluating hypotheses inferred by HypotheSAEs against the reference topics on Wiki.

Source	Overall AUC	# Sig	Significant Hypotheses: Red: ↑ Republican ; Blue: ↑ Democrat	Sep.	AUC
HypotheSAEs	0.702	15	*contains the phrase 'I ask unanimous consent'*	0.30	0.53
			mentions hearings conducted by Senate Committees or Subcommittees	0.18	0.52
			mentions scheduling or details about Senate votes or amendments	0.17	0.55
			discusses illegal immigration and its associated implications	0.15	0.51
			mentions victories or successes in military or political contexts	0.15	0.51
			mentions freedom or liberty	0.06	0.50
			discusses economic growth or growth rates	0.01	0.50
			discusses government spending or budgetary issues	-0.09	0.54
			mentions the need for reform or proposes reforms	-0.17	0.58
			mentions key figures related to the civil rights movement	-0.22	0.51
			criticizes government policies or actions	-0.29	0.63
			mentions the U.S. national debt or raising the debt limit	-0.40	0.51
			criticizes tax breaks or advantages for the wealthy	-0.44	0.52
			criticizes Republican leadership or policies	-0.45	0.60
			criticizes the administration's handling of the Iraq war	-0.45	0.52
BERTopic	0.636	10	*Senate procedural requests and adjournment schedules*	0.39	0.53
			Requests for unanimous consent to authorize committee meetings	0.32	0.53
			procedural discussions and scheduling of votes	0.20	0.56
			discusses immigration and border security	0.13	0.51
			mentions individuals with professional titles or roles	0.08	0.53
			debates on earmarks and federal spending processes	0.05	0.50
			discusses the Darfur conflict and international intervention	-0.30	0.50
			Congressional oversight of defense contracting in Iraq	-0.33	0.50
			discusses Asian Pacific American communities	-0.36	0.50
			Critiques Republican majority leadership	-0.46	0.55
NLParam	0.650	8	*expresses strong patriotism*	0.09	0.51
			contains specific legislative references	0.08	0.53
			mentions specific individuals or organizations	-0.03	0.52
			includes numerical data	-0.08	0.53
			addresses a critical national issue	-0.14	0.56
			expresses strong support or opposition	-0.15	0.58
			advocates for specific groups or communities	-0.17	0.55
			employs emotional or dramatic language	-0.23	0.58
HypoGeniC	0.675	5	*focuses on national security and immigration enforcement*	0.18	0.51
			discusses social issues such as healthcare and civil rights	-0.22	0.58
			mentions of government inefficiencies or calls for reform	-0.27	0.62
			advocates for civil liberties and critiques government overreach	-0.29	0.57
			emphasizes the need for government accountability	-0.37	0.62

Table 5. Congress hypotheses, for each method, with $p < 0.05$ after Bonferroni correction in a multivariate regression. The 'Sep.' column shows the separation score: $E[Y|\hat{Z} = 1] - E[Y|\hat{Z} = 0]$, where positive scores predict Republican speech and negative scores predict Democratic speech. The 'AUC' column provides univariate AUCs for individual hypotheses. 'Overall AUC' uses all hypotheses, including insignificant ones.

Source	Overall R^2	# Sig	Significant Hypotheses: Green: ↑ rating ; Red: ↓ rating	Sep.	R^2
HypotheSAEs	0.755	15	*expresses extreme enthusiasm or excitement with phrases like ’holy fuck’, ’can’t wait to go back’, or extensive use of exclamation points*	1.45	0.23
			uses exclamation marks enthusiastically to convey positive sentiment	1.41	0.23
			mentions anticipation or plans to return to the restaurant or order again	1.20	0.17
			explicitly states that a food item or restaurant is the best, often using superlative language	1.10	0.07
			mentions mediocrity or averageness of food or experience	-1.84	0.31
			mentions issues related to food safety, hygiene, or improper food handling	-1.98	0.15
			mentions issues with restaurant staff communication or behavior	-2.23	0.44
			mentions dissatisfaction or criticism of service, staff, or management	-2.25	0.61
			mentions food poisoning or symptoms of foodborne illness	-2.32	0.02
			mentions rude or aggressive behavior from restaurant staff	-2.37	0.30
			mentions dissatisfaction with restaurant policies, rules, or restrictions	-2.37	0.53
			criticizes the behavior or attitude of staff	-2.40	0.50
			complains about rude or unprofessional behavior from staff or management	-2.42	0.46
			mentions dishonest or unethical business practices by the establishment	-2.48	0.17
			describes a specific instance of poor customer service	-2.50	0.50
BERTopic	0.647	12	*praises the combination of excellent food quality and friendly, attentive service*	1.61	0.33
			describes detailed experiences with food quality, preparation, and flavor nuances	0.46	0.03
			criticizes the freshness, quality, or preparation of sushi and fish dishes	-1.77	0.07
			describes issues with restaurant closures or inaccurate operating hours	-1.77	0.10
			complains about uncleanliness of restaurant spaces	-1.91	0.05
			criticizes the quality and authenticity of Chinese dishes	-1.94	0.06
			criticizes Italian dishes, particularly pasta and sauces	-1.94	0.04
			complains about issues with pizza orders	-2.08	0.16
			criticizes breakfast food quality, portion sizes, and service experience	-2.15	0.37
			complains about issues with delivery orders	-2.20	0.25
			complains about poor service related to waiting times and staff behavior	-2.31	0.49
			complains about rude or unprofessional behavior from bartenders or bar staff	-2.31	0.31
NLParam	0.583	11	*uses enthusiastic or excessively positive language*	1.90	0.47
			describes exceptional food quality	1.65	0.35
			describes a positive recurring experience	1.45	0.26
			recommends the restaurant to others	1.29	0.18
			uses exclamation marks to express enthusiasm	1.22	0.17
			mentions attentive or friendly staff	1.01	0.12
			expresses enthusiasm for the ambiance or atmosphere	0.92	0.07
			mentions thoughtful accommodation of customer preferences	0.79	0.01
			describes generosity in portion sizes	0.69	0.02
			comments on portion size	0.02	0.00
			mentions long wait times to receive service	-1.70	0.15
HypoGeniC	0.775	12	*highlights a pleasant atmosphere or enjoyable dining experience*	2.40	0.66
			expresses a willingness to return or recommend the restaurant to others	2.13	0.55
			expresses a desire to return or recommends the place to others	1.89	0.45
			describes food as delicious or highlights specific dishes with positive adjectives	1.83	0.42
			notes issues with food quality, such as being cold or poorly cooked	-1.87	0.29
			includes complaints about long wait times for food or service	-1.94	0.30
			expresses disappointment with the quality of food compared to expectations	-2.17	0.49
			mentions poor service or unresponsive staff during the dining experience	-2.36	0.51
			complains about poor customer service or rude staff	-2.41	0.50
			complains about poor service or unhelpful staff	-2.42	0.60
			expresses disappointment with the overall dining experience	-2.42	0.67
			expresses disappointment with the overall experience or ambiance	-2.43	0.67

Table 6. Yelp hypotheses, for each method, with $p < 0.05$ after Bonferroni correction in a multivariate regression. ‘Sep.’ is the separation score: $E[Y|\hat{Z} = 1] - E[Y|\hat{Z} = 0]$, where positive scores predict higher restaurant ratings. The ‘R^2’ column provides univariate R^2 values of individual hypotheses. ‘Overall R^2’ uses all 20 hypotheses, including insignificant ones. Hypotheses for all methods are abbreviated.

Method	Total Count	Avg. Predictiveness	Avg. Fidelity
SAE	**45 / 60**	**0.717**	**0.836**
Embedding	30 / 60	0.706	0.545
PCA	23 / 60	0.697	0.547
Bottleneck	27 / 60	0.699	0.640

Table 7. Experiments testing ablations of the SAE on the three real-world datasets: HEADLINES, YELP, CONGRESS. Using the SAE yields many more significant hypotheses, slightly more predictive hypotheses, and much higher fidelity interpretations (i.e., interpretations that better explain the underlying neuron activations). The count of significant hypotheses is aggregated across the three datasets. Predictiveness is averaged across the three datasets. Fidelity is averaged across the 60 selected neurons (3 datasets, 20 neurons selected per dataset).

A. Data Preprocessing

WIKI. We download the dataset as processed by Pham et al. (2024); the data are derived from the WikiText corpus assembled by Merity et al. (2016). The articles are categorized by Wikipedia editors; each article has a supercategory, category, subcategory (e.g., `Media and Drama > Television > The Simpsons Episodes`). We focus on recovering article subcategories, since these are the most specific and therefore most difficult to fully recover. After filtering out duplicates and infrequent ($<$100 articles) subtopics, the dataset contains 11,979 items spanning 69 subcategories. For the WIKI-5 dataset, we set the ground-truth topics to the 5 most common subcategories. That is, we label all 1,771 articles (14.8%) in these subcategories as positives, while all other article subcategories are negatives. The WIKI-15 dataset is constructed similarly using the 15 most common subcategories (4,190 positives, 35.0%).

BILLS. We also download BILLS from Pham et al. (2024); the dataset was originally assembled by Hoyle et al. (2022) using GovTrack[6]. The bills are from the 110th-114th U.S. Congresses (Jan 2007 - Jan 2017). Each bill has a topic and subtopic. After filtering out very short bill texts (which may not have been properly transcribed), duplicate bill texts, bills with no subtopic, and bills with infrequent ($<$100) subtopics, there are 20,834 items spanning 70 subtopics. We label the 5,038 bills (24.2%) with the 5 most common subtopics as positives, and all others as negatives.

HEADLINES. We use the HEADLINES dataset as collected and released by the Upworthy Research Archive (Matias et al., 2021). All of the data is derived from web traffic to Upworthy.com, a digital media platform known for its attention-grabbing articles. The dataset consists of thousands of A/B tests, where users were randomly shown one of several possible headline variations for the same underlying article. Each headline is linked to a corresponding number of impressions and clicks, allowing researchers to estimate the causal effect of headline text on *click-rate* (clicks/impressions, also referred to as clickthrough-rate or engagement).

To preprocess the data, we start by downloading all publicly available data from the Upworthy archive[7], which includes clicks and impressions for 122,565 distinct headlines (we remove 28K rows from a problematic data range as recently reported by the dataset authors[8]). We then group headlines which have the same `test_id` (signifying that those headlines were randomized against one another). There are 14,128 such groups with at least two headlines. Following Zhou et al. (2024), we take the headlines with max and min click-rates in each group, and design a binary classification task to predict which headline in the pair received higher engagement. That is, we construct tuples (headline A, headline B, label), where headline A and headline B are randomly shuffled, and the label is 1 if headline A received higher click-rate than headline B.

Following Batista & Ross (2024), we take specific care to prevent train-holdout leakage on this dataset. Specifically, the same headline texts may be used in multiple randomized experiments (i.e., the same headline may occur in different `test_id`'s). We ensure that no identical headlines will appear in both the train split and the heldout split; we encourage future researchers to follow this practice, since splitting on `test_id` alone does not fully prevent headline leakage.

To handle the pairwise design, we train the SAE on the embeddings of all unique headlines in the training set, and then compute the activation matrices Z^A_{SAE} and Z^B_{SAE} for all of the headline A's and headline B's respectively. We select predictive

[6] https://www.govtrack.us/congress/bills/

[7] https://upworthy.natematias.com/index

[8] https://upworthy.natematias.com/2024-06-upworthy-archive-update.html

neurons by identifying columns in the matrix difference $\Delta Z_{\text{SAE}} := Z_{\text{SAE}}^{A} - Z_{\text{SAE}}^{B}$ which are predictive of $\mathbf{y}$.

YELP. We download the Yelp Open Dataset[9], and we filter to the 4.72M reviews where the business contains the 'Restaurant' tag. From this, we randomly sample training, validation, and heldout sets (200K, 10K, and 10K respectively).

CONGRESS. We download all congressional speech transcripts from the 109th U.S. Congress (Gentzkow & Shapiro, 2010)[10], which met from January 2005 to January 2007. We filter speeches which are very short (less than 250 characters) and include only speeches from Republican or Democrat speakers; the dataset is roughly balanced (51.7% Republican). If a speech is longer than ten sentences, we split it into ten sentence chunks, and exclude any chunks beyond the first five (to avoid over-representation of specific speeches). For the heldout set, we restrict to one chunk per speech to ensure that our hypotheses generalize to a diverse corpus. Ultimately, we use 114K speech chunks for training, 16K for validation, and 12K for heldout eval. If a speech has multiple excerpts, we ensure that all of them are included in the same split to avoid leakage.

B. Hyperparameters

B.1. Training sparse autoencoders

We train k-sparse autoencoders, where each forward pass masks all neurons besides the top-k activating ones (Makhzani & Frey, 2014; Gao et al., 2024). The most influential hyperparameters, which we tune for each dataset, are M and k. We choose (M, k) by maximizing validation performance. Specifically, for each combination of (M, k), we evaluate validation AUC (or R^2 on YELP) after fitting an L_1-regularized predictor with exactly $H \ll M$ nonzero coefficients. We constrain M and k to powers of two to tractably limit the search space.

These parameters, broadly, control the level of granularity of the concepts learned by each SAE neuron: M is the total number of concepts which can be learned across the entire dataset, and k is the number of concepts which can be used to represent each instance. For intuition, assume that with $(M, k) = (16, 4)$ on YELP, the SAE learns a single neuron that fires when reviews mention *price*; with $(M, k) = (32, 4)$, the SAE may learn separate neurons for reviews which mention *high prices* vs. *low prices* (described as "feature splitting" in Bricken et al. (2023)); with $(M, k) = (32, 8)$, the SAE may learn to represent more niche features altogether, such as the *number of exclamation marks*. In general, increasing either parameter yields more granular features, at the risk of producing some neurons that are uninterpretable or redundant.

In some cases we would like features at multiple levels of granularity: for example, we may want both the general feature "mentions prices" and the more specific feature "mentions that the cocktails were cheap during happy hour", but training a single large SAE (e.g. $(1024, 32)$) may only produce features as specific as the latter. A simple solution is to train multiple SAEs of different sizes, concatenate their activation matrices, and select features from any of them. We find that this approach is helpful (improves validation AUC) on the HEADLINES dataset. Ultimately, we use the following values:

- WIKI-5: $(32, 4)$.
- WIKI-15 & BILLS: $(64, 4)$.
- HEADLINES: $(256, 8)$; $(32, 4)$.
- YELP: $(1024, 32)$.
- CONGRESS: $(4096, 32)$.

Following prior work, we add several optimizations to improve SAE training. Following best practices based on results from Bricken et al. (2023), we (i) tie the weights of the pre-encoder bias and the decoder bias, $b_{\text{pre}} = b_{\text{dec}}$; (ii) initialize b_{pre} to the geometric median of the training set; (iii) initialize $W_{\text{dec}} = W_{\text{enc}}^T$; and (iv) force all decoder rows to have unit norm.

To minimize the number of "dead" latent neurons in the SAE—neurons which are never selected by the TopK filter—we use an auxiliary loss (Gao et al., 2024; O'Neill et al., 2024). The auxiliary loss term provides nonzero gradients for up to

[9] https://business.yelp.com/data/resources/open-dataset/

[10] https://data.stanford.edu/congress_text

k_{aux} neurons which were not active in the last several forward passes. We fit these neurons to the residual reconstruction error left by the TopK neurons. That is, let z_i^{AuxK} consist of the activations of the top k_{aux} dead neurons in z_i; then

$$\mathcal{L}_{\text{aux}} = ||W_{\text{dec}} z_i^{\text{AuxK}} - (e_i - \hat{e}_i)||_2^2,$$

and the full loss, then, is $\mathcal{L} = \mathcal{L}_{\text{TopK}} + w_{\text{aux}} \mathcal{L}_{\text{AuxK}}$.

Hyperparameters for the auxiliary loss and other training hyperparameters are given below (identical for all experiments):

- $\boldsymbol{k}_{\text{aux}}$: $2 \cdot k$; that is, up to $2 \cdot k$ dead neurons are used to fit the TopK reconstruction error.
- **Dead neuron threshold steps**: 256. This is the number of consecutive steps for which a neuron was not selected by TopK to be considered "dead."
- **Auxilliary loss coefficient** $w_{\textbf{aux}}$: $1/32$. This parameter gives the weight of the auxiliary reconstruction's loss.
- **Batch size**: 512; **learning rate**: 5e-4; **gradient clipping threshold**: 1.0.
- **Epochs**: Up to 200, with early stopping after 5 epochs of validation loss not decreasing.

B.2. Interpreting neurons with LLMs

There are several parameters for neuron interpretation which we fix across experiments:

- **Interpretation model**: GPT-4o (version 2024-11-20).
- **Temperature**: 0.7.
- **Number of highly-activating examples**: 10.
- **Number of weakly-activating examples**: 10.
- **Maximum word count per example**: 256 (examples longer than this are truncated).
- **Number of candidate interpretations**: 3 (we choose the highest-fidelity interpretation out of 3 candidates).
- **Number of samples to evaluate fidelity**: 200.

In Appendix C, we describe experiments which justify these choices: they either improve fidelity or have no effect compared to alternate choices. We also provide the full interpretation prompt in Figure 5.

There is one hyperparameter which we adjust per dataset: the **bin percentiles** from which we sample highly-activating and weakly-activating examples for the neuron interpretation prompt. By default, highly-activating texts are sampled from the top decile of positive activations $[90, 100]$, and weakly-activating texts are sampled from the bottom decile of positive activations $[0, 10]$. This works well in general, especially for larger datasets (YELP, CONGRESS). For smaller datasets like WIKI, the $[90, 100]$ bin produces interpretations which are too specific (e.g., "articles about Napoleon", even if the neuron fires on articles about European leaders more generally). Note that this requires a judgment call: what is "too specific" depends on practitioner needs. For example, on WIKI, we choose bins such that the resulting interpretations are roughly as granular as the topics in the dataset[11]. We ultimately choose the following percentile bins for highly-activating texts:

- WIKI-5, WIKI-15, HEADLINES: $[80, 100]$.
- BILLS: $[50, 100]$.
- YELP, CONGRESS: $[90, 100]$.

[11] Choosing this parameter requires similar decision-making to choosing the minimum cluster size in BERTOPIC.

Choosing the **number of candidate interpretations** depends on budget. In Appendix C we find that selecting the interpretation with highest fidelity from a pool of 3 candidates increases the resulting fidelity modestly, yet statistically significantly; we therefore use **3 candidate interpretations** for all experiments. However, we also ablate this step—that is, generating a single candidate interpretation and using it directly, without computing fidelity. On CONGRESS (we did not test on the other datasets), the resulting hypotheses are of similar quality: 14/20 are significant, with AUC 0.70, compared to 15/20 significant and AUC 0.70 when using 3 candidate interpretations[12]. This variant of our method, SAE-NOVAL, costs as little as BERTOPIC (Table 3).

C. Supporting Experiments

Our final neuron interpretation procedure for our main experiments is described in §4, with hyperparameters provided in Appendix B.2. Here, we describe several experiments to justify our procedure. The section is organized as follows. In C.1, we show that our metric of interpretation fidelity—the F1 score comparing concept annotations against neuron activations—is a useful selection metric to improve downstream hypothesis quality. In C.2, we test several different interpretation strategies and hyperparameters and measure their impacts on fidelity. In C.3, we analyze how much prediction signal is lost at each stage of our method, shedding light on how future work can improve hypothesis quality.

C.1. Higher-fidelity interpretations improve hypotheses.

To test whether fidelity is a useful metric to select from different neuron interpretations, we conduct the following experiment on the Yelp dataset:

1. Train an SAE (Step 1), and identify the top-20 predictive neurons via Lasso (Step 2).
2. Generate 10 candidate interpretations (at temperature 0.7, and with different random samples in the interpretation prompt) for each neuron using the hyperparameters in B.2 (Step 3).
3. Compute the *fidelity* for each candidate interpretation: that is, binarizing 100 examples in the top decile of the neuron's positive activations as positives, and 100 examples in the bottom decile of the neuron's positive activations as negatives, compute the F1 score between the GPT-4o-mini annotations according to the interpretation against the binarized neuron activations.
4. Store two sets of neuron interpretations: one set includes the interpretations with the best F1 score for each neuron (high-fidelity), and the other set includes the interpretations with the worst F1 score for each neuron (low-fidelity).
5. Compute annotations on the full heldout set (10K examples) for the high-fidelity hypotheses and the low-fidelity hypotheses, and compare how predictive they are as measured by R^2.

This procedure yields three interesting results. First, there is enough variation in the 10 candidate interpretations such that the best and worst F1 scores are substantially different: the median best F1 is 0.79, while the median worst F1 is 0.53. Second, we find that the high-fidelity set is indeed more predictive: in a multivariate OLS, the high-fidelity set achieves an R^2 of 0.76, compared to 0.61 for the low-fidelity set.

Third, most importantly, using a neuron's high-fidelity interpretation results in a more predictive hypothesis, on average. Figure 3 shows that, for 17 out of 20 neurons, moving from the low-fidelity interpretation to the high-fidelity interpretation increases the R^2 of the resulting hypothesis (p-value, paired t-test: 0.002). For example, for one neuron, the high-fidelity interpretation is "*mentions disputes or issues related to billing or refunds*" (F1: 0.79), while the low-fidelity interpretation is more generically "*mentions an attempt by the restaurant or management to address or respond to a problem or complaint*." The former hypothesis achieves heldout $R^2 = 0.10$ with restaurant rating, while the latter is not significantly correlated with rating ($R^2 = 0.003$). These experiments show that **fidelity—as measured via F1 score—is a useful heuristic to choose between multiple candidate neuron interpretations**, when our goal is to produce predictive hypotheses.

Finally, following prior work (Bills et al., 2023), we also tested another metric of fidelity: the R^2 between the concept annotations and the neuron activations across 100 top-activating samples and 100 random samples. Choosing the top candidate interpretation based on this alternative fidelity metric did not produce hypotheses that were significantly more or less predictive than using our F1 fidelity metric.

[12] To illustrate this point, we tested solely on CONGRESS; we expect similarly modest differences on other datasets.

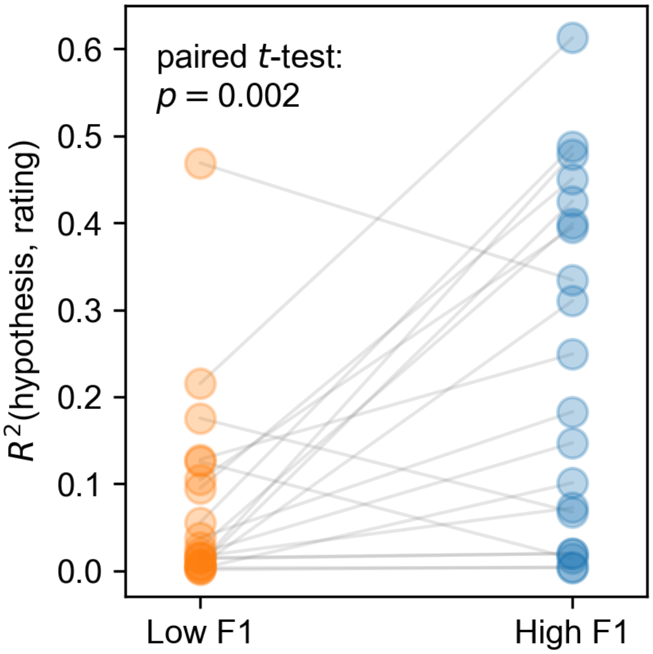

Figure 3. For 17 of the top 20 neurons on YELP, a higher-fidelity neuron interpretation results in a more predictive hypothesis. Fidelity is measured by the F1 score between the concept annotations and the neuron activations on a sample of 100 highly-activating and 100 weakly-activating examples. We generate 10 candidate interpretations for each neuron, and compare the interpretations with best- and worst-F1. Predictiveness is measured by computing the R^2 between the concept annotations and Y (restaurant rating) for 10K heldout examples.

C.2. Effects of the neuron interpretation procedure on fidelity.

Having established that improving interpretation fidelity improves downstream predictiveness, we search across different hyperparameters for neuron interpretation to maximize fidelity. Our final procedure is to generate 3 candidate interpretations for each neuron by prompting GPT-4o with 10 examples in the top decile of positive activations vs. 10 examples in the bottom decile of positive activations. We use the highest fidelity interpretation out of these 3. There are two sources of non-determinism: each candidate interpretation is generated with a different random seed to sample examples, and we use an LLM temperature of 0.7.

To compare this strategy against other strategies—or to compare strategy A and strategy B more generally—we compute interpretations, and their F1 scores, for the top 100 neurons on YELP under both strategies separately. We run a t-test comparing the pairs of resulting F1 scores to test whether one strategy produces consistently higher F1 scores than the other.

We observe the following:

- **Number of candidate interpretations:** Relative to baseline (generating 3 candidate interpretations and taking the one with highest F1 score), generating only a single candidate interpretation is significantly worse (-0.04). Generating 5 candidate interpretations improves significantly on 3 (+0.01 F1, paired t-test $p = 0.003$), but the improvement is modest and results in a slower, more expensive method. Therefore, we use only 3 candidate interpretations.
- **Top-random vs. high-weak sampling strategy:** Prior work (e.g. Templeton et al. (2024); Bills et al. (2023)) interprets SAE neurons by prompting with the top-activating texts and random texts. Relative to our baseline (prompting with highly-activating texts, but not necessarily top-activating; and weakly-activating texts instead of random), this alternate approach has no significant effect on F1 score, though the interpretations tend to be more specific: that is, they have higher precision and lower recall. This is because a neuron's absolute top-activating examples generally share a more specific characteristic than examples sampled randomly from the top decile. Though neither strategy wins clearly in terms of F1, we choose high-weak sampling for our experiments because, empirically, top-random produces hypotheses which can be *too* specific (thereby lowering predictiveness); this is discussed further in B.2. **Our code defaults to top-random** for

simplicity, but practitioners should treat sampling strategy as a tunable hyperparameter that depends on desired hypothesis granularity.

- **Continuous vs. binary prompt structure:** Prior work (e.g. Zhong et al. (2024)) prompts the interpreter LLM with a sorted list of texts and their continuous activations. Relative to our baseline (showing separate lists of positive samples and negative samples), this significantly lowers F1 score (-0.06), so we use the binary prompt structure.

- **Chain-of-thought:** Prior work (e.g. O'Neill et al. (2024)) uses a chain-of-thought reasoning prompt to interpret neurons, where the LLM is encouraged to iteratively discover a concept which all of the positive samples contain and none of the negative samples contain. Relative to baseline, we found that CoT has an insignificant but weakly negative effect (-0.03 mean F1), and also sometimes fails to output any interpretation. However, this is still a promising line for future work: it's possible that while GPT-4o does not reason well, better models or reasoning-specific models (OpenAI o1, DeepSeek R1, *etc.*) will produce higher fidelity labels with chain-of-thought.

- **Sample count:** Relative to baseline (prompting with 20 total samples, 10 highly-activating and 10 weakly-activating), prompting with 10 total samples lowers average F1 score by 0.03. Prompting with 50 samples has no effect, so we use 20 to favor shorter prompts.

- **Temperature:** Relative to baseline (temperature 0.7), using a lower (0.0) or higher (1.0) temperature has an insignificant but weakly negative effect (-0.01 in mean F1), so we use 0.7.

C.3. Localizing losses in predictive performance to different steps of the method.

HypotheSAEs starts from embeddings and ultimately produces hypotheses, after three steps. It is worth asking—how does predictive performance change at each step of our method? Answering this question points to possible improvements: if the SAE's full hidden representation is much less predictive of restaurant ratings than the input text embeddings, perhaps we need a larger (more expressive) SAE, or more data to train the SAE's representations; meanwhile, if the largest performance loss comes from translating neurons into natural language hypotheses, perhaps we need a stronger LLM for interpretation.

Figure 4 displays this experiment. At each stage, we fit a logistic or linear regression on the training set to predict the labels on Headlines, Yelp, and Congress. The four stages are: (i) text embeddings (length-1536 vectors of floats); (ii) the full SAE activation matrix, Z_{SAE} (length-288, 1024, and 4096 vectors of floats for the three tasks, respectively); (iii) the activations of the top-20 selected SAE neurons (length-20 vectors of floats); and (iv) the annotations from the interpreted hypotheses (length-20 vectors of binary 0/1 values).

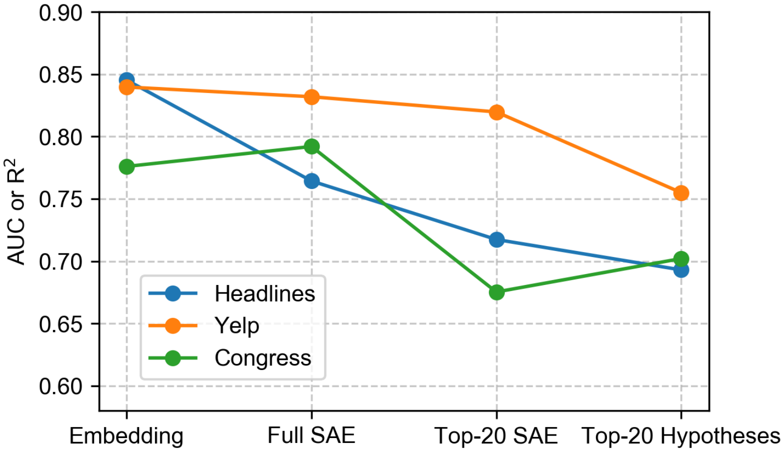

Figure 4. Performance losses at each step of the HypotheSAEs pipeline. The leftmost point shows the performance of training a supervised single-layer model to predict the outcome using text embeddings; the second point shows the supervised performance using the full SAE representation; the third point shows the supervised performance using the top-20 SAE neurons; the rightmost point shows the supervised performance using the top-20 hypothesis annotations after neuron interpretation.

The trends depend on the dataset. On Headlines, we find that the performance drops most at the first stage (-0.08 AUC): the full SAE is significantly less expressive than the original embeddings. This could suggest using a larger SAE, but empirically we find that increasing the SAE size does not improve AUC, perhaps because there is not enough data ($\sim$14K headlines) to make use of more parameters. On YELP, we find that performance is mostly retained until neuron interpration, at which point R^2 drops by 0.06. This suggests some combination of: our neuron interpretations are imperfect, GPT-4o-mini is not annotating the hypotheses faithfully, or binary annotations are not expressive enough to capture the neuron activations (which are continuous). Finally, on CONGRESS, the only drop comes from using the top-20 neurons to predict the label instead of the full SAE representation (4096 neurons). This implies that 20 features aren't expressive enough; or, more optimistically, that there are more than 20 distinct hypotheses to discover on this dataset (see §6.4). Remarkably, there are no losses in the interpretation step on the CONGRESS dataset, suggesting that on this dataset our interpretations are very high-fidelity. Ultimately, this analysis can be a useful diagnostic tool to direct one's efforts when trying to improve hypothesis quality: losses from embedding $\Rightarrow$ full SAE suggest that the dataset and/or the SAE are too small; losses from full SAE $\Rightarrow$ top-H SAE suggest that H is too small; losses from top-H SAE $\Rightarrow$ top-H hypotheses suggest that the interpretation step can be improved. Meanwhile, if the top-H hypotheses achieve similar AUC to the underlying embeddings, this implies that there may be no more "juice to squeeze" in the dataset.

D. Hyperparameters for Baseline Methods

BERTOPIC (Grootendorst, 2022). To run BERTOPIC[13], we feed in the same OpenAI embeddings that we use for HYPOTHESAES, and otherwise use default choices for the algorithm: UMAP for dimensionality reduction, HDBSCAN for clustering, and c-TF-IDF to compute the top words associated with each topic. We tune the minimum cluster size hyperparameter in BERTOPIC, which controls topic granularity. The default value is 10, which often produces topics which are too granular (e.g., relative to the underlying granularity of the subtopics on WIKI and BILLS). We test all values in $\{10, 20, 50, 100, 200, 500\}$. We choose this parameter by maximizing the validation performance of the L_1-regularized predictor which uses H features; this is identical to how we choose hyperparameters for HYPOTHESAES. We use default values of all other hyperparameters, which is standard in prior work (e.g., Pham et al. (2024)). After running BERTOPIC, we label topics using the prompt Figure 8.

NLPARAM (Zhong et al., 2024). We run NLPARAM using the publicly available implementation[14]. We set the number of candidate labels per bottleneck layer neuron to 3 (the same as our method), and we use the same text embeddings as our method (OpenAI's text-3-embedding-small). Zhong et al. (2024) release two different prompts, one for "simple" hypotheses and one for "complex" hypotheses; following their work, we use the "simple" prompt for our 3 synthetic tasks and the "complex" prompt for our 3 real-world tasks. For all other hyperparameters, we use default values.

HYPOGENIC (Zhou et al., 2024). We run HYPOGENIC using the publicly available implementation[15]. We use the default parameters as defined in examples/generation.py. Following the format in their example tasks, we write custom prompts in a consistent format describing each our of tasks. The method uses the same LLM for hypothesis generation and hypothesis scoring, and we run all experiments using GPT-4o-mini (the default model in their code repository, and also the model that we use for scoring hypotheses in HYPOTHESAES).

E. Qualitative Evaluation

To confirm that our hypotheses pass a "sniff test," we recruit three computational social science researchers to perform a qualitative evaluation. These researchers are not authors on our study. We collect ratings for two attributes, following Lam et al. (2024): (1) *helpfulness* and (2) *interpretability*. We explain these to the raters as follows:

1. Helpful: Does this hypothesis help you understand what predicts the target variable? If you were studying this dataset, is this a hypothesis you would want to and could explore further? Rate 1 if yes, 0 if no or only a little.
2. Interpretable: When you read the hypothesis, is it clear what it means? Can you easily and unambiguously apply it to new examples? Rate 1 if yes, 0 if no or it would often be subjective.

We collect 3 annotations for each of the significant hypotheses outputted by any method on the HEADLINES and CONGRESS

[13] https://github.com/MaartenGr/BERTopic
[14] https://github.com/ruiqi-zhong/nlparam
[15] https://github.com/ChicagoHAI/hypothesis-generation

datasets. (We removed the YELP dataset to reduce annotator load.) Hypotheses are shown to annotators in random order, i.e., they were not grouped by method. We aggregate labels by taking the majority label.

F. Proofs

Proof of Proposition 3.1. Set $p_{ab} := \Pr[\hat{Z} = a, Z = b]$ and $y_{ab} := \mathbb{E}[Y|\hat{Z} = a, Z = b]$ for $a, b \in \{0, 1\}$. Then

$$\mathbb{E}[Y|\hat{Z} = 1] = \frac{y_{11}p_{11} + y_{10}p_{10}}{p_{11} + p_{10}} \tag{8}$$

$$\mathbb{E}[Y|Z = 1] = \frac{y_{11}p_{11} + y_{01}p_{01}}{p_{11} + p_{01}}. \tag{9}$$

Therefore,

$$\mathbb{E}[Y|\hat{Z} = 1] - \mathbb{E}[Y|Z = 1] = \frac{y_{11}p_{11} + y_{10}p_{10}}{p_{11} + p_{10}} - \frac{y_{11}p_{11} + y_{01}p_{01}}{p_{11} + p_{01}} \tag{10}$$

$$= y_{11}\left(\frac{p_{11}}{p_{11} + p_{10}} - \frac{p_{11}}{p_{11} + p_{01}}\right) + \frac{y_{10}p_{10}}{p_{11} + p_{10}} - \frac{y_{01}p_{01}}{p_{11} + p_{01}} \tag{11}$$

$$= y_{11}\left(\frac{p_{01}}{p_{11} + p_{01}} - \frac{p_{10}}{p_{11} + p_{10}}\right) + \frac{y_{10}p_{10}}{p_{11} + p_{10}} - \frac{y_{01}p_{01}}{p_{11} + p_{01}} \tag{12}$$

Since $y_{10} \leq 1$ and $y_{01} \geq 0$, (12) is bounded from above by

$$y_{11}\left(\frac{p_{01}}{p_{11} + p_{01}} - \frac{p_{10}}{p_{11} + p_{10}}\right) + \frac{p_{10}}{p_{11} + p_{10}}. \tag{13}$$

If $\frac{p_{01}}{p_{11}+p_{01}} - \frac{p_{10}}{p_{11}+p_{10}} \geq 0$, (13) is bounded from above by

$$\left(\frac{p_{01}}{p_{11} + p_{01}} - \frac{p_{10}}{p_{11} + p_{10}}\right) + \frac{p_{10}}{p_{11} + p_{10}} = \frac{p_{01}}{p_{11} + p_{01}}, \tag{14}$$

since $y_{11} \leq 1$. Otherwise, (13) is bounded from above by

$$\frac{p_{10}}{p_{11} + p_{10}}. \tag{15}$$

Also note that (12) is bounded from below by

$$y_{11}\left(\frac{p_{11}}{p_{11} + p_{10}} - \frac{p_{11}}{p_{11} + p_{01}}\right) - \frac{p_{01}}{p_{11} + p_{10}} = y_{11}\left(\frac{p_{01}}{p_{11} + p_{01}} - \frac{p_{10}}{p_{11} + p_{10}}\right) - \frac{p_{01}}{p_{11} + p_{10}}. \tag{16}$$

If $\frac{p_{01}}{p_{11}+p_{01}} - \frac{p_{10}}{p_{11}+p_{10}} \leq 0$, (16) is bounded from below by

$$\left(\frac{p_{01}}{p_{11} + p_{01}} - \frac{p_{10}}{p_{11} + p_{10}}\right) - \frac{p_{01}}{p_{11} + p_{10}} = -\frac{p_{10}}{p_{11} + p_{10}}, \tag{17}$$

since $y_{11} \leq 1$. Otherwise, (16) is bounded from below by

$$-\frac{p_{01}}{p_{11} + p_{01}}. \tag{18}$$

Therefore,

$$|\mathbb{E}[Y|\hat{Z} = 1] - \mathbb{E}[Y|Z = 1]| \leq \max\left\{\frac{p_{10}}{p_{11} + p_{10}}, \frac{p_{01}}{p_{11} + p_{01}}\right\}. \tag{19}$$

Analogously, we can bound $|\mathbb{E}[Y|\hat{Z} = 0] - \mathbb{E}[Y|Z = 0]|$ by

$$\max\left\{\frac{p_{10}}{p_{00} + p_{10}}, \frac{p_{01}}{p_{00} + p_{01}}\right\}. \tag{20}$$

Notice that

$$\frac{p_{10}}{p_{00}+p_{10}} = \frac{p_{10}}{p_{11}+p_{10}} \cdot \frac{p_{11}+p_{10}}{p_{00}+p_{10}} \tag{21}$$

$$= \frac{p_{10}}{p_{11}+p_{10}} \cdot \frac{\Pr[\hat{Z}=1]}{\Pr[Z=0]} \tag{22}$$

Similarly,

$$\frac{p_{01}}{p_{00}+p_{01}} = \frac{p_{01}}{p_{11}+p_{01}} \cdot \frac{p_{11}+p_{01}}{p_{00}+p_{01}} \tag{23}$$

$$= \frac{p_{01}}{p_{11}+p_{01}} \cdot \frac{\Pr[Z=1]}{\Pr[\hat{Z}=0]} \tag{24}$$

Set $p := \min\{\Pr[\hat{Z}=0], \Pr[Z=0]\}$. Then $\Pr[\hat{Z}=0], \Pr[Z=0] \geq p$ and $\Pr[\hat{Z}=1], \Pr[Z=1] \leq 1-p$

$$\frac{\Pr[\hat{Z}=1]}{\Pr[Z=0]}, \frac{\Pr[Z=1]}{\Pr[\hat{Z}=0]} \leq \frac{1-p}{p}. \tag{25}$$

It follows that

$$\left|S(\hat{Z}) - S(Z)\right| = \left|(\mathbb{E}[Y|\hat{Z}=1] - \mathbb{E}[Y|\hat{Z}=0]) - (\mathbb{E}[Y|Z=1] - \mathbb{E}[Y|Z=0])\right| \tag{26}$$

is at most

$$\left(1 + \frac{1-p}{p}\right) \max\left\{\frac{p_{10}}{p_{11}+p_{10}}, \frac{p_{01}}{p_{11}+p_{01}}\right\} = \frac{1}{p} \max\left\{\frac{p_{10}}{p_{11}+p_{10}}, \frac{p_{01}}{p_{11}+p_{01}}\right\}. \tag{27}$$

The result follows by noting that $p := \min\{\Pr[\hat{Z}=0], \Pr[Z=0]\}$ and $\frac{p_{10}}{p_{11}+p_{10}}, \frac{p_{01}}{p_{11}+p_{01}} = 1 - \text{ recall}, 1 - \text{ precision}.$ □

G. Prompts

You are a machine learning researcher who has trained a neural network on a text dataset. You are trying to understand what text features cause a specific neuron in the neural network to fire.

You are given two sets of SAMPLES: POSITIVE SAMPLES that strongly activate the neuron, and NEGATIVE SAMPLES from the same distribution that do not activate the neuron. Your goal is to identify a feature that is present in the positive samples but absent in the negative samples.
Example features could be:
- “uses multiple adjectives to describe colors”
- “describes a patient experiencing seizures or epilepsy”
- “contains multiple single-digit numbers”

POSITIVE SAMPLES:
————————-
{positive_samples}
————————-

NEGATIVE SAMPLES:
————————-
{negative_samples}
————————-

Rules about the feature you identify:
- The feature should be objective, focusing on concrete attributes rather than abstract concepts.
- The feature should be present in the positive samples and absent in the negative samples. Do not output a generic feature which also appears in negative samples.
- The feature should be as specific as possible, while still applying to all of the positive samples. For example, if all of the positive samples mention Golden or Labrador retrievers, then the feature should be “mentions retriever dogs”, not “mentions dogs” or “mentions Golden retrievers”.

Do not output anything else. Your response should be formatted exactly as shown in the examples above. Please suggest exactly one description, starting with “-” and surrounded by quotes “”. Your response is:
- “

Figure 5. The prompt we use to interpret SAE neurons. The positive samples are texts that strongly activate the neuron, while the negative samples are texts that weakly activate the neuron. The “example features” section can be edited to produce features in a different format.

```
Is text_a and text_b similar in meaning? Respond with yes, related, or no.

Here are a few examples.

Example 1:
text_a: has a topic of protecting the environment
text_b: has a topic of environmental protection and sustainability
output: yes

Example 2:
text_a: has a language of German
text_b: has a language of Deutsch
output: yes

Example 3:
text_a: has a topic of the relation between political figures
text_b: has a topic of international diplomacy
output: related

Example 4:
text_a: has a topic of the sports
text_b: has a topic of sports team recruiting new members
output: related

Example 5:
text_a: has a named language of Korean
text_b: uses archaic and poetic diction
output: no

Example 6:
text_a: has a named language of Korean
text_b: has a named language of Japanese
output: no

Example 7:
text_a: describes an important 20th century historical event
text_b: describes a 20th century European politician
output: no

Target:
text_a: {text_a}
text_b: {text_b}
output:
```

Figure 6. The prompt we use to evaluate surface similarity between the inferred hypotheses and reference topics. The prompt is lightly edited from Zhong et al. (2024). The few-shot examples demonstrate three levels of similarity: equivalent meaning (yes), related concepts (related), and unrelated concepts (no).

Check whether the TEXT satisfies a PROPERTY. Respond with Yes or No with an explanation that discusses the evidence from the TEXT (at most a sentence). When uncertain, output No.

Example 1:
PROPERTY: “mentions a natural scene.”
TEXT: “I love the way the sun sets in the evening.”
Output: Yes. “Sun sets” are clearly natural scenes.

Example 2:
PROPERTY: “writes in a 1st person perspective.”
TEXT: “Jacob is smart.”
Output: No. This text is written in a 3rd person perspective.

Example 3:
PROPERTY: “is better than group B.”
TEXT: “I also need to buy a chair.”
Output: No. It is unclear what the PROPERTY means (e.g., what does group B mean?) and doesn’t seem related to the text.

Example 4:
PROPERTY: “mentions that the breakfast is good on the airline.”
TEXT: “The airline staff was really nice! Enjoyable flight.”
Output: No. Although the text appreciates the flight experience, it DOES NOT mention about the breakfast.

Example 5:
PROPERTY: “appreciates the writing style of the author.”
TEXT: “The paper absolutely sucks because its underlying logic is wrong. However, the presentation of the paper is clear and the use of language is really impressive.”
Output: Yes. Although the text dislikes the paper, it says “the presentation of the paper is clear”, so it DOES like the writing style.

Example 6:
PROPERTY: “has a formal style; specifically, the language in the text is relatively formal, complex and academic. For example, ’represent whom and which”’
TEXT: “investigates formation of nominalization”
Output: Yes. “formation” and “nominalization” are abstract and complex nouns.

Example 7:
PROPERTY: “refers to historical dates; specifically, there are references to years or specific dates in the text. For example, ’Obama was born on August 4, 1961.”’
TEXT: “A member of the Democratic Party, he was the first African-American president of the United States.”
Output: No. The text does not mention date.

Now complete the following example - Respond with Yes or No with an explanation that discusses the evidence from the TEXT. When uncertain, output No.

PROPERTY: {hypothesis}
TEXT: {text}
Output:

Figure 7. The prompt we use to compute hypothesis annotations with GPT-4o-mini to evaluate hypothesis quality. The prompt is lightly edited from Zhong et al. (2024).

You are a machine learning researcher analyzing clusters from a topic model. You need to identify the key distinguishing features of documents in a specific topic cluster.
Example features could be:
- “uses multiple adjectives to describe colors”
- “describes a patient experiencing seizures or epilepsy”
- “contains multiple single-digit numbers”

These are the top keywords describing the topic: {topic_keywords}.

You will also see a sample of documents belonging to this topic and a random sample of documents from other topics from the same dataset.

TOPIC DOCUMENTS (a sample of documents belonging to this topic):
——————-
{topic_documents}
——————-

RANDOM DOCUMENTS (a sample of documents from other topics):
——————-
{random_documents}
——————-

Your goal is to create a short, specific label that captures what makes the topic documents different from the contrasting documents.

The label should be:
1. As specific as possible, while still applying to all topic documents
2. Focused on concrete features rather than abstract concepts
3. Based on characteristics present in the topic documents but not in the contrasting documents

Do not output anything else. Your response should be a single topic label starting with “-” and surrounded by quotes “”. Your response is:
- “

Figure 8. The prompt we use to generate interpretable topic labels for BERTopic. The LLM is prompted with the keywords that distinguish the topic from others, and also with two sets of documents: 10 documents belonging to the target topic, and a random sample of 10 documents from other topics. This labeling approach is inspired by Grootendorst (2024), but we designed the prompt ourselves (intending it to be similar, where possible, to our neuron interpretation prompt).