

Paper Review : 2025 Summer Lab Seminar

# Scaling Monosemanticity : Extracting Interpretable Features from Claude 3 Sonnet

---

Adly Templeton\*, Tom Conerly\*, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, Tom Henighan

Anthropic Inc. | Transformer Circuits Thread; Circuit Tracing | 2024.05

**Yejin Yoon**

Paper Review : 2025 Summer Lab Seminar

Towards Monosemanticity: ~ ↴

# Scaling Monosemanticity : Extracting Interpretable Features from Claude 3 Sonnet

---

what “Circuit Tracing” aims to do ↴

Adly Templeton\*, Tom Conerly\*, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, Tom Henighan

Anthropic Inc. | Transformer Circuits Thread; Circuit Tracing | 2024.05

Yejin Yoon

Paper Review : 2025 Summer Lab Seminar

Towards Monosemanticity: ~ 

# Scaling Monosemanticity

## : Extracting Interpretable Features from Claude 3 Sonnet

---

what “Circuit Tracing” aims to do 

Adly Templeton\*, Tom Conerly\*, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, Tom Henighan

Anthropic Inc. | Transformer Circuits Thread; Circuit Tracing | 2024.05

Yejin Yoon

# Pre-Requisite

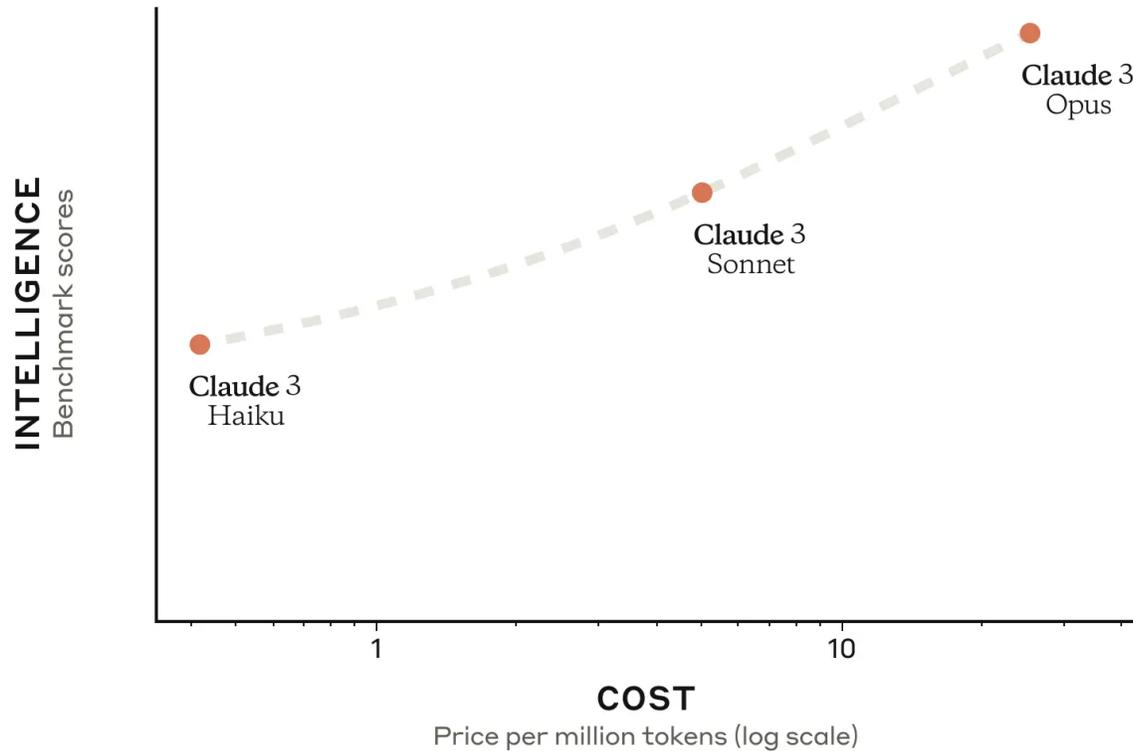
# Claude 3 Sonnet

# SAE

# TL; DR

# Claude 3 Family

“Claude 3 Family: Introducing Claude 3 Haiku, Sonnet, and Opus” (Anthropic, 2024.03)

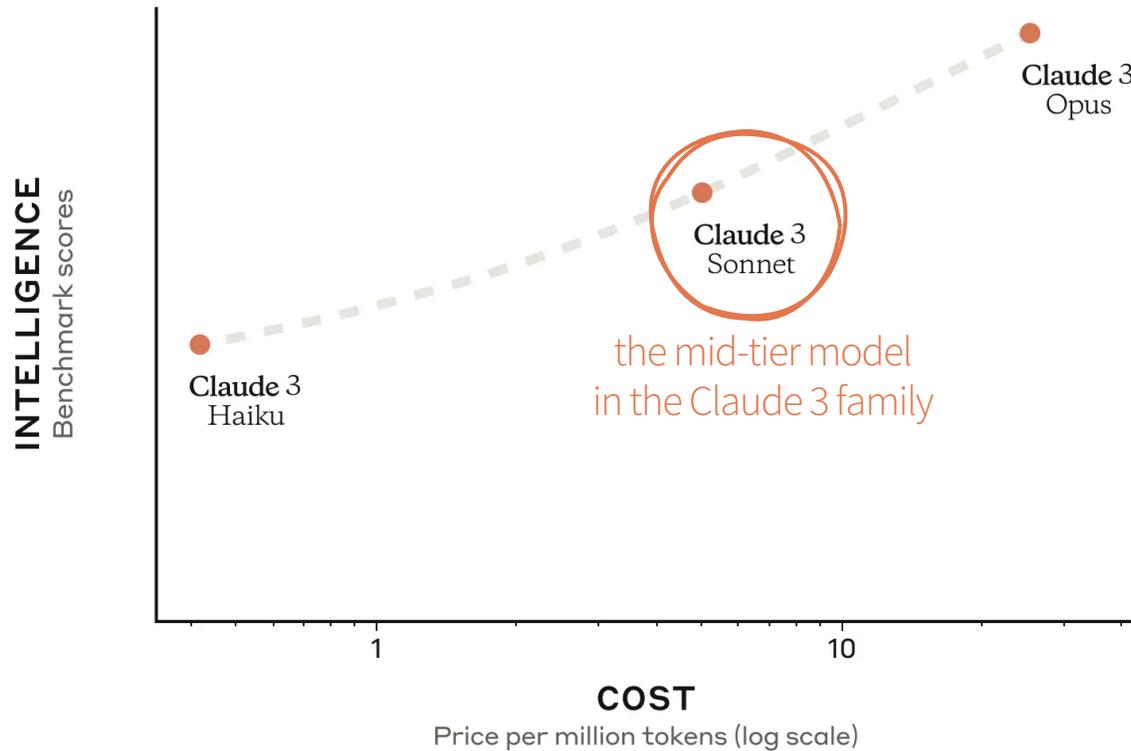


	Claude 3 Opus	Claude 3 Sonnet	Claude 3 Haiku	GPT-4	GPT-3.5	Gemini 1.0 Ultra	Gemini 1.0 Pro
Undergraduate level knowledge <i>MMLU</i>	86.8% 5-shot	79.0% 5-shot	75.2% 5-shot	86.4% 5-shot	70.0% 5-shot	83.7% 5-shot	71.8% 5-shot
Graduate level reasoning <i>GPQA, Diamond</i>	50.4% 0-shot CoT	40.4% 0-shot CoT	33.3% 0-shot CoT	35.7% 0-shot CoT	28.1% 0-shot CoT	—	—
Grade school math <i>GSM8K</i>	95.0% 0-shot CoT	92.3% 0-shot CoT	88.9% 0-shot CoT	92.0% 5-shot CoT	57.1% 5-shot	94.4% Maj1@32	86.5% Maj1@32
Math problem-solving <i>MATH</i>	60.1% 0-shot CoT	43.1% 0-shot CoT	38.9% 0-shot CoT	52.9% 4-shot	34.1% 4-shot	53.2% 4-shot	32.6% 4-shot
Multilingual math <i>MGSM</i>	90.7% 0-shot	83.5% 0-shot	75.1% 0-shot	74.5% 8-shot	—	79.0% 8-shot	63.5% 8-shot
Code <i>HumanEval</i>	84.9% 0-shot	73.0% 0-shot	75.9% 0-shot	67.0% 0-shot	48.1% 0-shot	74.4% 0-shot	67.7% 0-shot
Reasoning over text <i>DROP, F1 score</i>	83.1 3-shot	78.9 3-shot	78.4 3-shot	80.9 3-shot	64.1 3-shot	82.4 Variable shots	74.1 Variable shots
Mixed evaluations <i>BIG-Bench-Hard</i>	86.8% 3-shot CoT	82.9% 3-shot CoT	73.7% 3-shot CoT	83.1% 3-shot CoT	66.6% 3-shot CoT	83.6% 3-shot CoT	75.0% 3-shot CoT
Knowledge Q&A <i>ARC-Challenge</i>	96.4% 25-shot	93.2% 25-shot	89.2% 25-shot	96.3% 25-shot	85.2% 25-shot	—	—
Common Knowledge <i>HellaSwag</i>	95.4% 10-shot	89.0% 10-shot	85.9% 10-shot	95.3% 10-shot	85.5% 10-shot	87.8% 10-shot	84.7% 10-shot

**Claude 3 family**—Haiku, Sonnet, and Opus—highlighting major advances in LC understanding, vision capabilities, and model alignment

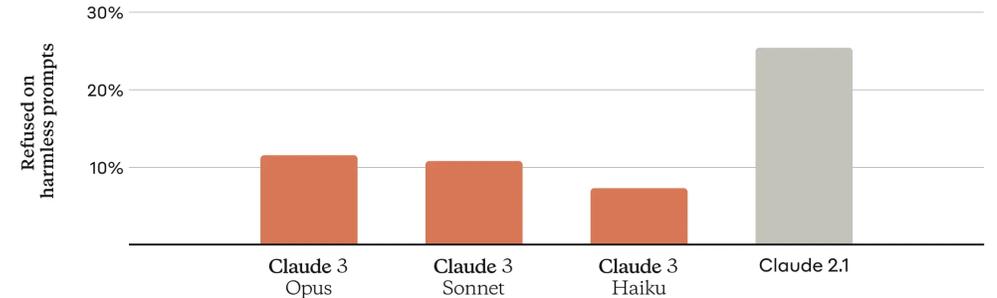
# Claude 3 Sonnet

“Claude 3 Family: Introducing Claude 3 Haiku, Sonnet, and Opus” (Anthropic, 2024.03)



	Claude 3 Opus	Claude 3 Sonnet	Claude 3 Haiku	GPT-4V	Gemini 1.0 Ultra	Gemini 1.0 Pro
Math & reasoning <i>MMMU (val)</i>	59.4%	53.1%	50.2%	56.8%	59.4%	47.9%
Document visual Q&A <i>ANLS score, test</i>	89.3%	89.5%	88.8%	88.4%	90.9%	88.1%
Math <i>MathVista (testmini)</i>	50.5% CoT	47.9% CoT	46.4% CoT	49.9%	53.0%	45.2%
Science diagrams <i>AI2D, test</i>	88.1%	88.7%	86.7%	78.2%	79.5%	73.9%
Chart Q&A <i>Relaxed accuracy (test)</i>	80.8% 0-shot CoT	81.1% 0-shot CoT	81.7% 0-shot CoT	78.5% 4-shot CoT	80.8%	74.1%

Incorrect refusals



**Claude 3 Sonnet** delivers strong performance at a lower cost compared to its peers —the ideal balance between intelligence and speed

# Claude 3 Sonnet

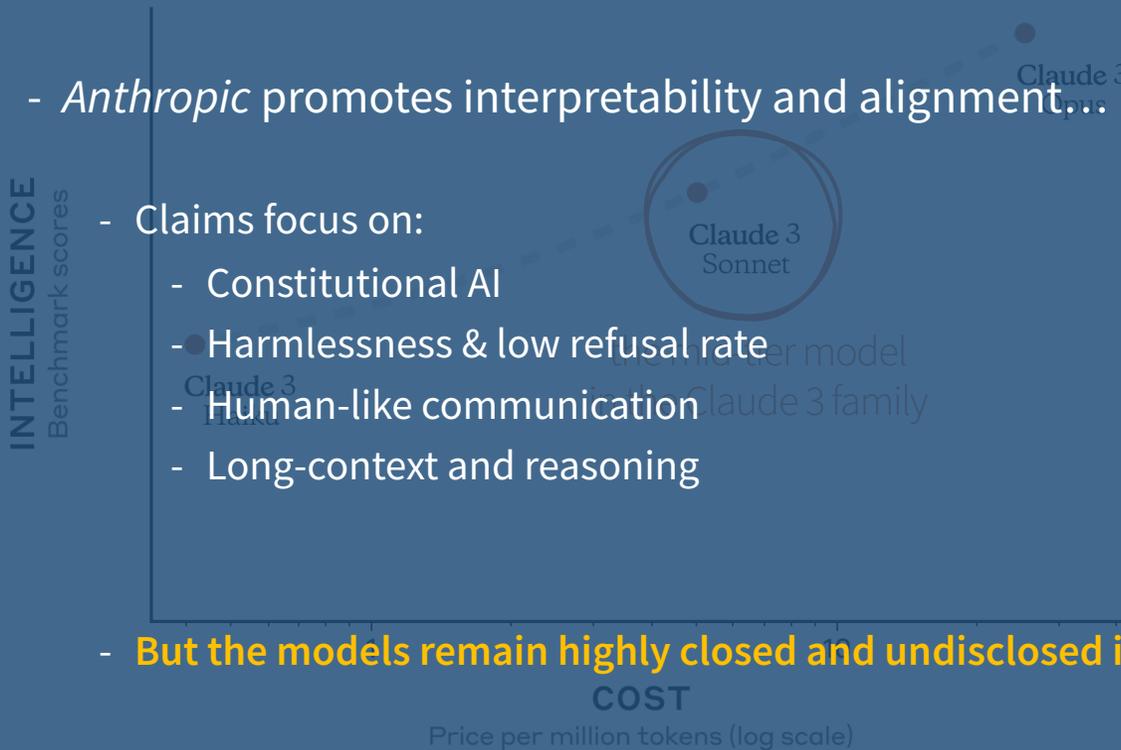
## • Interpretability Paradox : The Ironic Opacity of Claude

- Anthropic promotes interpretability and alignment...

- Claims focus on:

- Constitutional AI
- Harmlessness & low refusal rate
- Human-like communication
- Long-context and reasoning

- But the models remain highly closed and undisclosed in architecture and training details.



Component	Claude 3 Opus	Claude 3 Sonnet	Claude 3 Haiku	GPT-4V	Gemini 1.0 Ultra	Gemini 1.0 Pro
Math & reasoning MMMU (val)	59.4%	53.1%	50.2%	56.8%	59.4%	47.9%
Document visual Q&A ANLS score, test	89.5%	88.8%	88.1%	88.1%	88.1%	88.1%
Math MathVista (test)	47.9%	46.4%	45.2%	Not disclosed	45.2%	45.2%
Science dia AI2D, test	88.7%	86.7%	86.7%	Not disclosed (only estimated)	86.7%	86.7%
Chart Q&A Relaxed accuracy (test)	81.1%	81.7%	74.1%	Not disclosed	80.8%	74.1%
Training data composition	Not disclosed (estimated mix of public/private)					
Incorrect refusals	Not disclosed					
Prompt formatting	Not disclosed					
Fine-tuning procedure	Not disclosed					

Anthropic promotes interpretability, yet treats Claude models as black boxes.

→ Hinders trustworthy interpretability research and fair alignment comparisons ☹️

# Auto Encoder

## • What is an AutoEncoder?

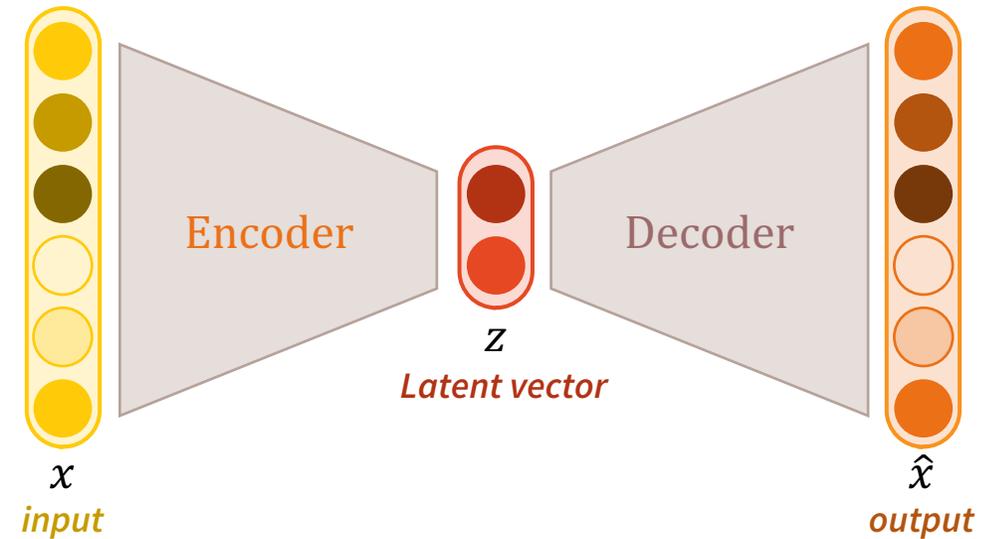
- A neural network that learns to **compress** and reconstruct input data.

$$x \xrightarrow{\text{Encoder}} z \xrightarrow{\text{Decoder}} \hat{x}$$

- **Encoder** : maps input  $x$  to latent vector  $z$ .  $f(x) \rightarrow z$
- **Decoder** : reconstructs  $\hat{x} \approx x$  from  $z$ .  $D \cdot z \rightarrow \hat{x}$
- Typically trained with:

$$\mathcal{L}_{\text{AE}} = ||x - \hat{x}||_2^2$$

- Learns low-dimensional representations
- Assumes data lies on a low-dimensional manifold



AE learns a **compact representation**.

# Sparse Auto Encoder

## • What is a Sparse AutoEncoder?

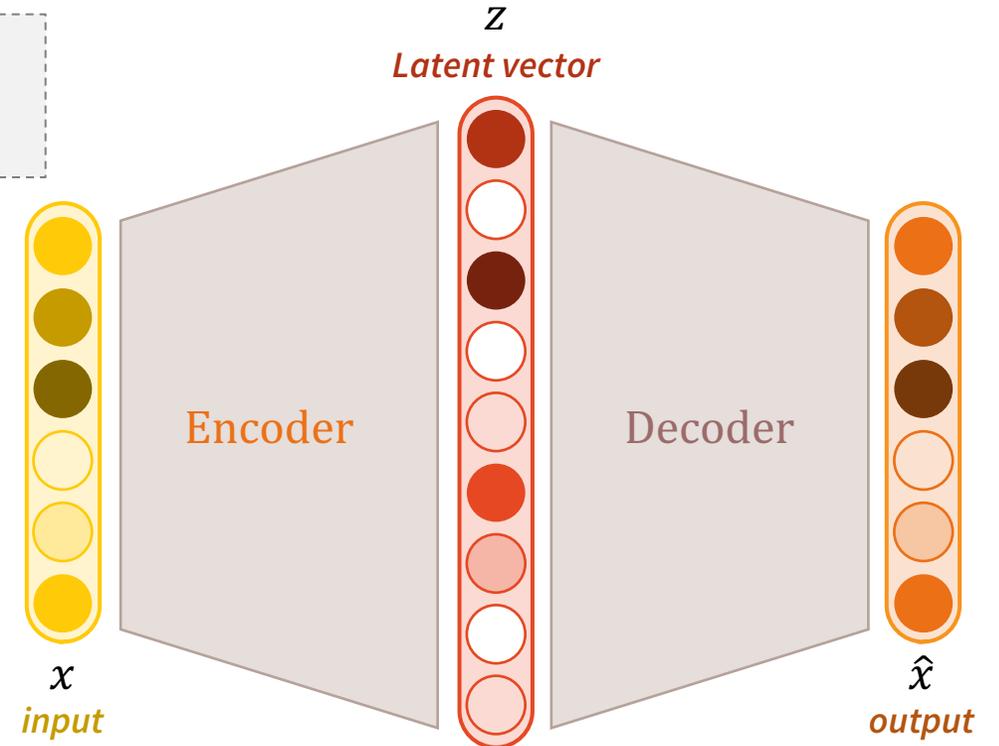
- A neural network that learns to **compress** and reconstruct input data.

$$x \xrightarrow{\text{Encoder}} z \xrightarrow{\text{Decoder}} \hat{x}$$

- **Encoder** : maps input  $x$  to latent vector  $z$ .  $f(x) \rightarrow z$
- **Decoder** : reconstructs  $\hat{x} \approx x$  from  $z$ .  $D \cdot z \rightarrow \hat{x}$
- Typically trained with:

$$\mathcal{L}_{\text{SAE}} = \underbrace{\|x - \hat{x}\|_2^2}_{\text{L2-term: Reconstruction}} + \underbrace{\lambda \|z\|_1}_{\text{L1-term: Sparsity Penalty}}$$

- use the **L1-norm** to induce sparsity in the latent features



AE learns a **compact representation**. SAE learns a **sparse, overcomplete representation**.

# Sparse Auto Encoder

- **Scaling Dictionary Learning to Claude 3 Sonnet**

- (High-level) goal: to decompose the activations of a model (Claude 3 Sonnet) into more interpretable pieces.

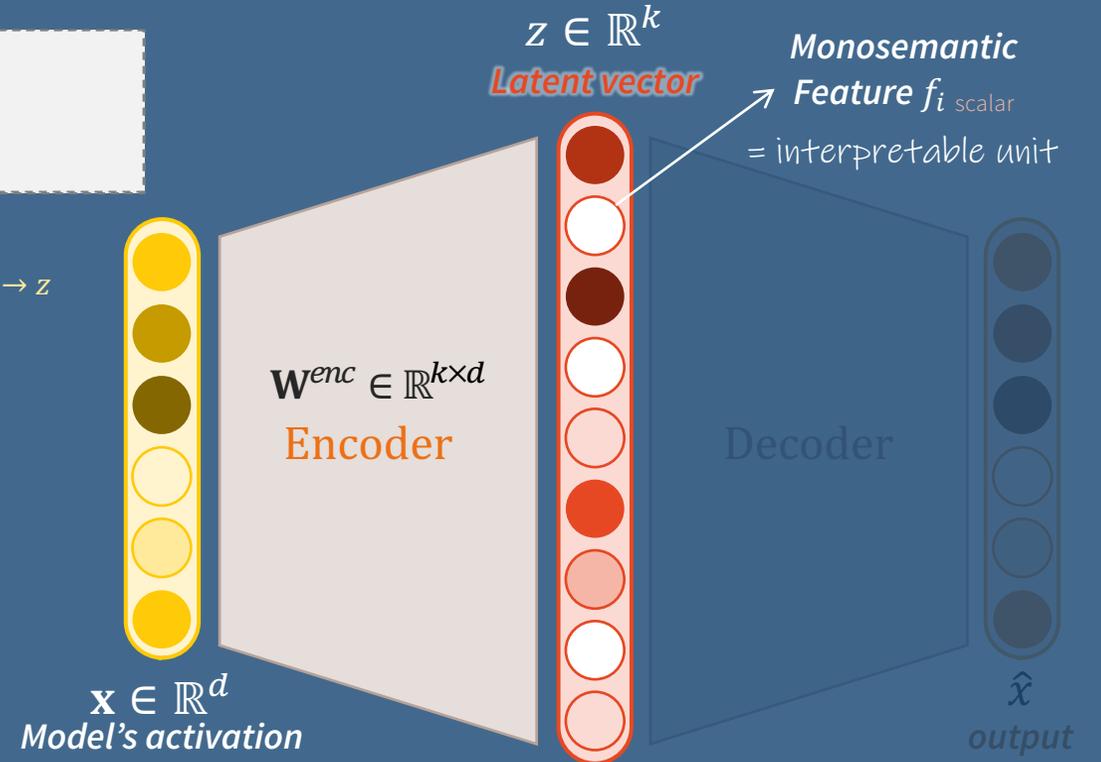
$$x \xrightarrow{f(x) \rightarrow z} z \xrightarrow{D \cdot z \rightarrow \hat{x}} \hat{x}$$

- **Encoder**: maps the activity to a higher-dimensional layer via a learned linear transformation followed by a ReLU nonlinearity. Latent vector  $f(x) \rightarrow z$

*\*encoder vector:  
a specific direction in feature space*

$$f_i(x) = \text{ReLU}(W_i^{enc} \cdot x + b_i)$$

nonlinearity      linear transformation



*Superposition hypothesis:*

“The model’s internal activations are **linear combinations** of many overlapping features.”

# Sparse Auto Encoder

## • Scaling Dictionary Learning to Claude 3 Sonnet

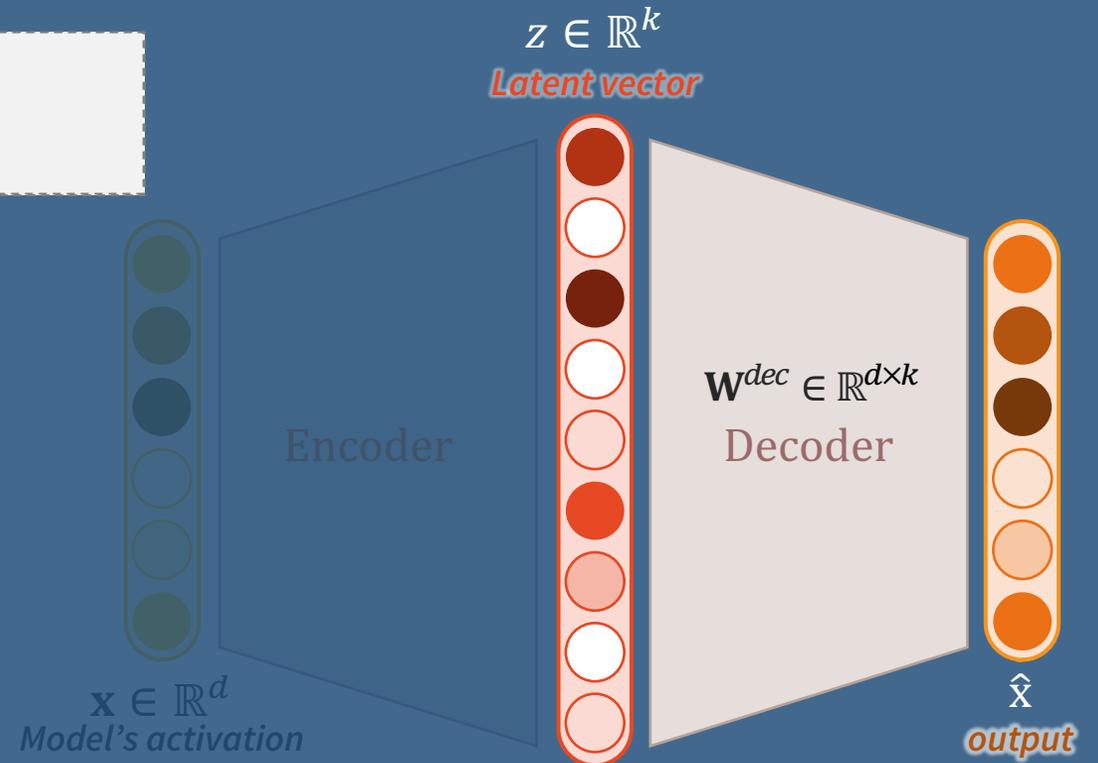
- (High-level) goal: to decompose the activations of a model (Claude 3 Sonnet) into more interpretable pieces.

$$\mathbf{x} \xrightarrow{f(\mathbf{x}) \rightarrow \mathbf{z}} \mathbf{z} \xrightarrow{D \cdot \mathbf{z} \rightarrow \hat{\mathbf{x}}} \hat{\mathbf{x}}$$

- Encoder:  $f_i(x) = \text{ReLU}(W_i^{enc} \cdot x + b_i)$   
nonlinearity linear transformation
- **Decoder**: reconstructs the model activations  
via a linear transformation of the feature activations.

$$\begin{aligned} \hat{\mathbf{x}} &= \mathbf{b} + \sum_i f_i(x) W_i^{dec} \quad \text{linear reconstruction} \\ &= f_1(x) \cdot w_1 + f_2(x) \cdot w_2 + \dots \end{aligned}$$

*\*decoder vector: activation basis*



*Superposition hypothesis:*

“The model’s internal activations are **linear combinations** of many overlapping features.”

# Sparse Auto Encoder

## • Scaling Dictionary Learning to Claude 3 Sonnet

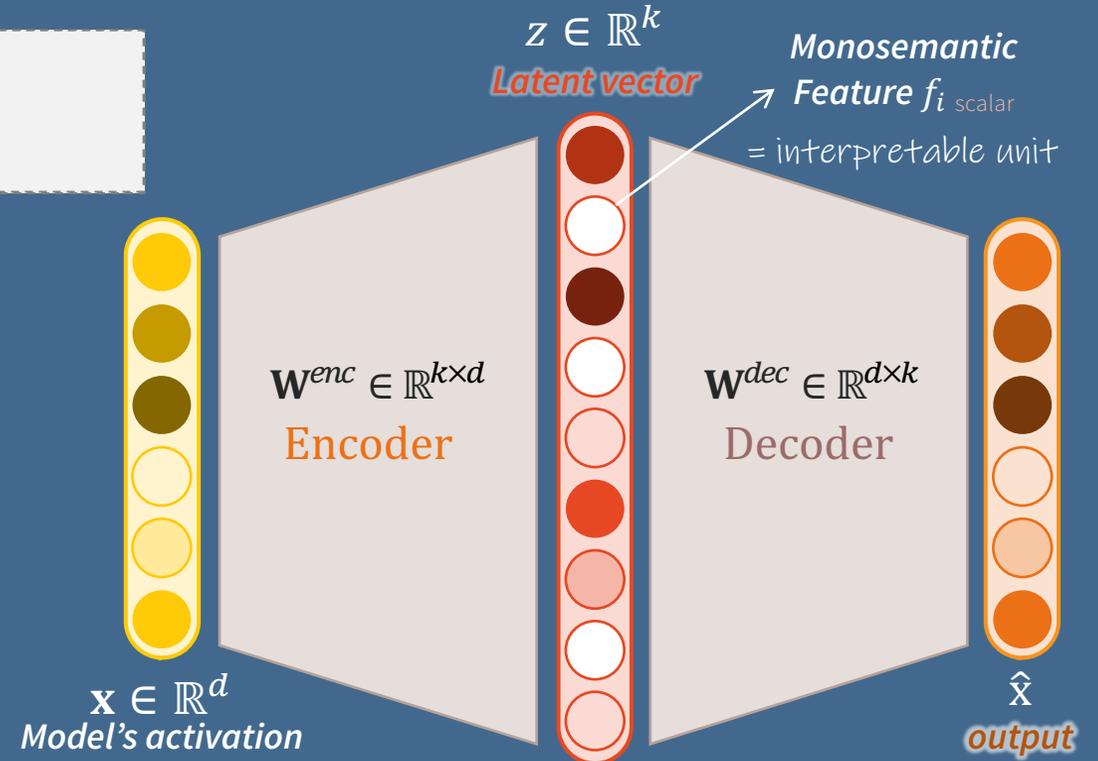
- (High-level) goal: to decompose the activations of a model (Claude 3 Sonnet) into more interpretable pieces.

$$\mathbf{x} \xrightarrow{f(\mathbf{x}) \rightarrow \mathbf{z}} \mathbf{z} \xrightarrow{D \cdot \mathbf{z} \rightarrow \hat{\mathbf{x}}} \hat{\mathbf{x}}$$

- Encoder:  $f_i(x) = \text{ReLU}(W_i^{enc} \cdot x + b_i)$   
nonlinearity linear transformation
- Decoder:  $\hat{x} = b + \sum_i f_i(x) W_i^{dec}$  linear reconstruction

- Trained with:

$$\mathcal{L}_{SAE} = \underbrace{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2}_{\text{L2-term: Reconstruction}} + \lambda \underbrace{\sum_i f_i(x) \cdot \|W_{:,i}^{dec}\|_2}_{\text{L1-term: Sparsity Penalty}}$$



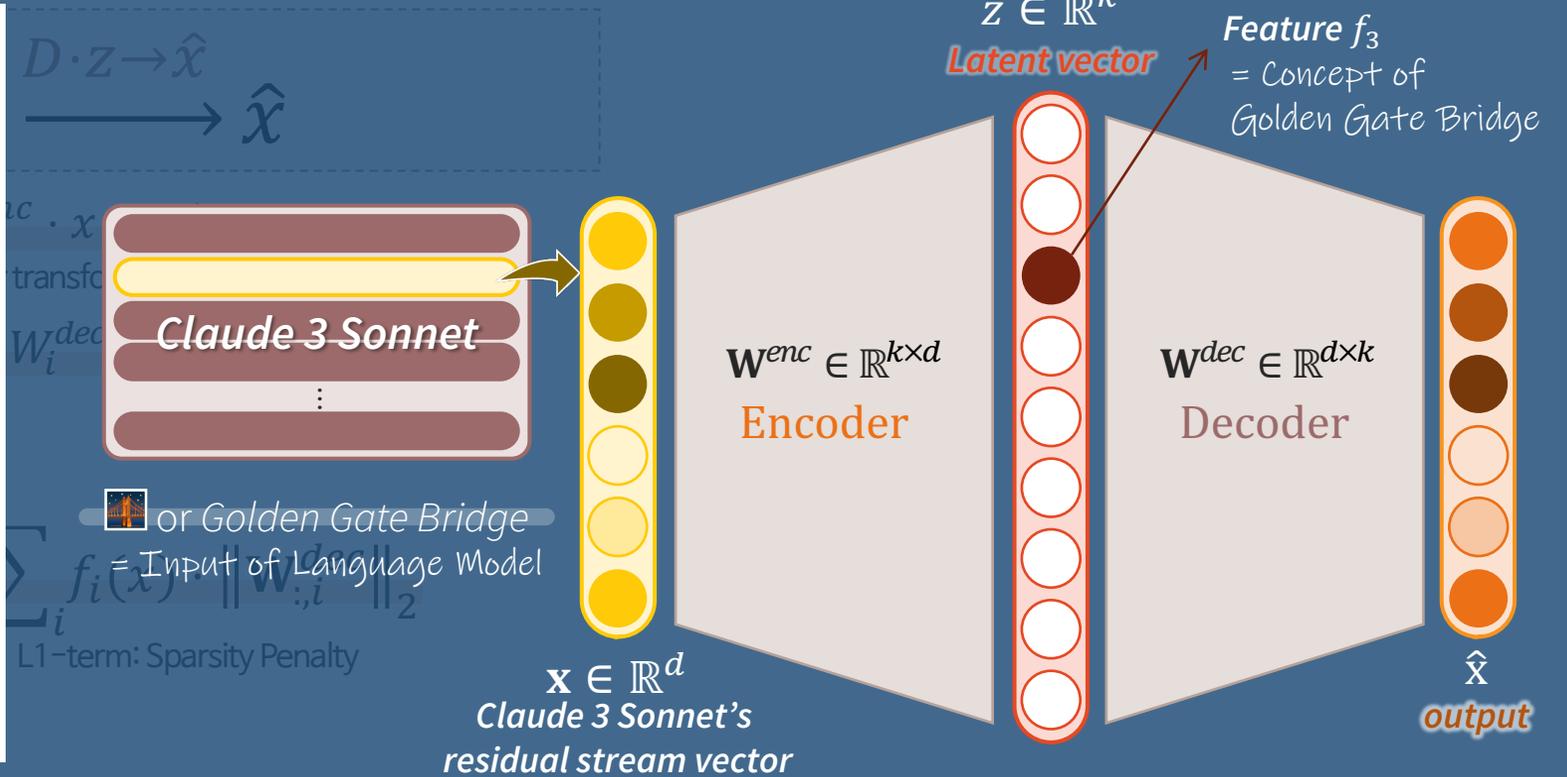
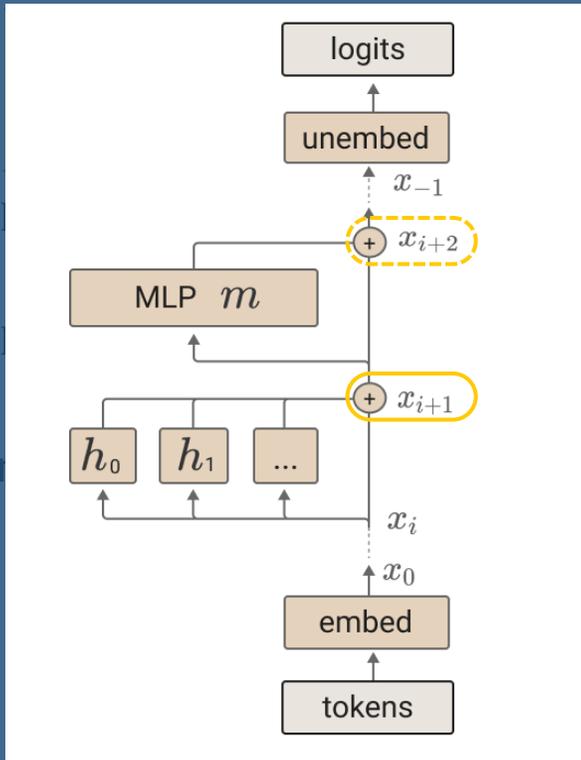
*Superposition hypothesis:*

“The model’s internal activations are **linear combinations** of many overlapping features.”

# Sparse Auto Encoder

- **Scaling Dictionary Learning to Claude 3 Sonnet**

- (High-level) goal: to decompose the activations of a model (Claude 3 Sonnet) into more interpretable pieces.



Focusing on applying SAEs to **residual stream activations** halfway through the model

# TL; DR

---

- **Background:**

- Prior work applied SAEs to small models, but it's unclear if this scales to frontier models like *Claude 3 Sonnet*.

- **Problem States:**

- Can we extract monosemantic, interpretable features from *Claude 3 Sonnet*?
- Does interpretability improve with scale?

- **Suggestions:**

1. Train SAEs on Claude 3 residual activations (mid-layer).
2. Use L1 regularization to ensure sparse activation (~80 of 16,384).
3. Evaluate feature interpretability via machine annotation.

- **Effects:**

- 5K+ features are clearly interpretable (e.g., “reversal,” “anger”).
- Sparse features help in probing, auditing, and concept discovery.
- SAE performance scales with compute (power law).

Scaling Monosemanticity

# Contents

## Scaling Dictionary Learning to Claude 3 Sonnet Assessing Feature Interpretability

- Four Examples of Interpretable Features
- Sophisticated Features
- Features vs. Neurons

## Feature Survey

- Exploring Feature Neighborhoods
- Feature Completeness
- Feature Categories

## Features as Computational Intermediates

- Example: Emotional Inferences
- Example: Multi-Step Inference

## Searching for Specific Features

### Safety-Relevant Features

- Safety-Relevant Code Features
- Bias Features
- Sycophancy Features
- Deception, Power-seeking and Manipulation-related Features
- Case Study: Detecting and Correcting Deception using Features
- Criminal or Dangerous Content Features
- Features Relating to the Model's Representation of Self
- Comparison to Other Approaches

## Discussion

## Related Work

## 1 Scaling Sparse Autoencoders to Large Models

*Can sparse autoencoders still work on production-scale models like Claude 3 Sonnet?*

## 2 Interpretable Features: Do They Really Exist?

*Can we find features with consistent and meaningful activations?*

## 3 The Structure of the Feature Space

*How are features organized semantically and geometrically?*

## 4 Features as Causal Units of Computation

*Do features actually influence the model's reasoning and outputs?*

## 5 Searching and Steering: Interacting with the Feature Space

*How can we discover and manipulate meaningful features on demand?*

## 6 Safety-Relevant Features and Model Risks

*What safety-relevant patterns are encoded in features?*

## 7 Discussion and Open Problems

*What are the current limitations and where do we go next?*

Scaling Monosemanticity

# Contents

## 1 **Scaling Sparse Autoencoders to Large Models**

*Can sparse autoencoders still work on production-scale models like Claude 3 Sonnet?*

## 2 **Interpretable Features: Do They Really Exist?**

*Can we find features with consistent and meaningful activations?*

## 3 **The Structure of the Feature Space**

*How are features organized semantically and geometrically?*

## 4 **Features as Causal Units of Computation**

*Do features actually influence the model's reasoning and outputs?*

## 5 **Searching and Steering: Interacting with the Feature Space**

*How can we discover and manipulate meaningful features on demand?*

## 6 **Safety-Relevant Features and Model Risks**

*What safety-relevant patterns are encoded in features?*

## 7 **Discussion and Open Problems**

*What are the current limitations and where do we go next?*

Paper Review : 2025 Summer Lab Seminar

# Scaling Monosemanticity : Extracting Interpretable Features from Claude 3 Sonnet

---

Adly Templeton\*, Tom Conerly\*, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, Tom Henighan

Anthropic Inc. | Transformer Circuits Thread; Circuit Tracing | 2024.05

**Yejin Yoon**

# Scaling SAEs to Large Models

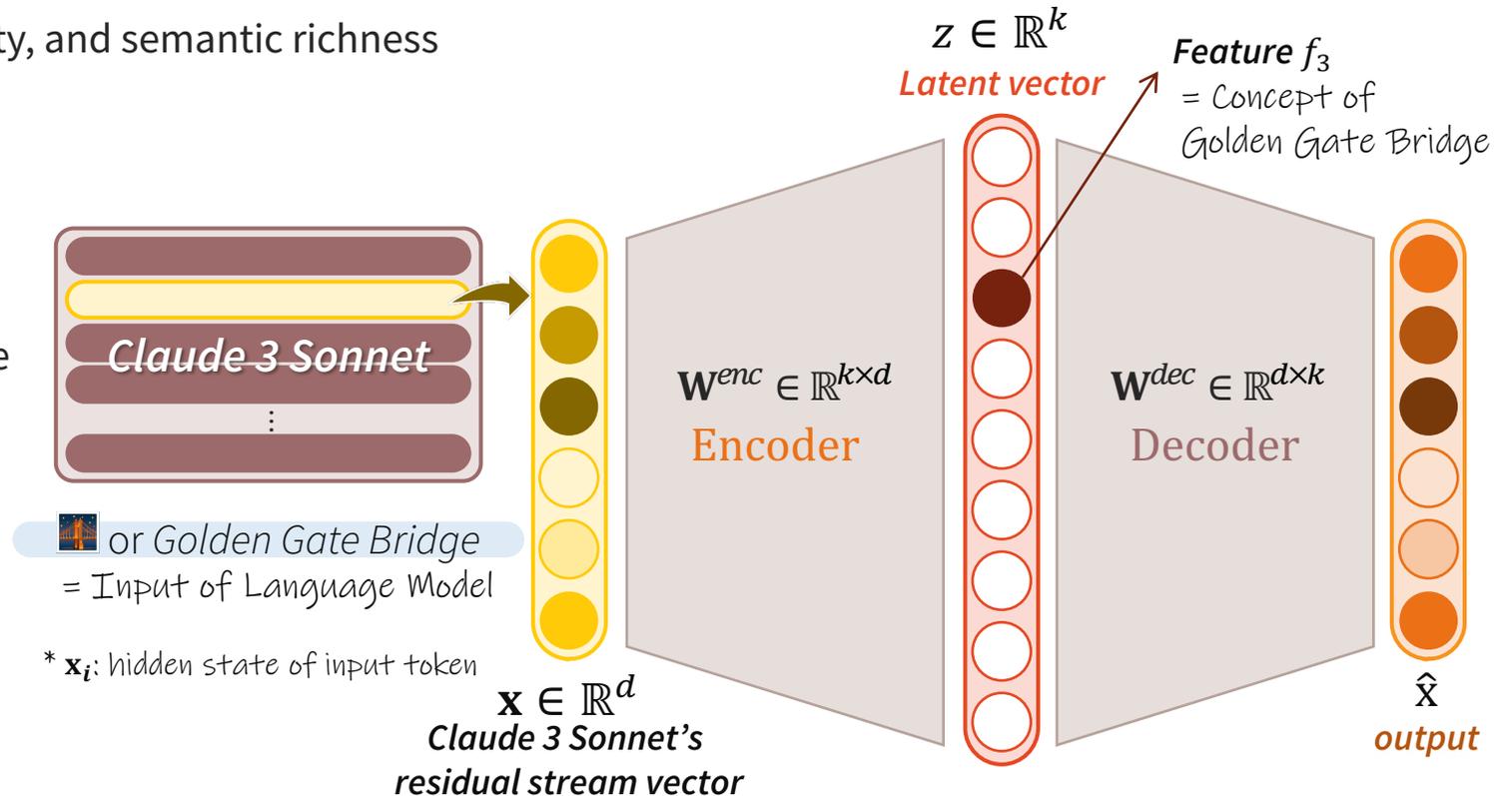
# SAE Experiments Setup

# Scaling Law

# SAE Experiments Setup

## • Scaling SAEs: Feature Utilization and Dead Units

- **Target:** Residual stream activations at the **middle layer**
  - Chosen for efficiency, interpretability, and semantic richness
- Trained SAE sizes ( $k$ ):
  - ~1M, ~4M, ~34M features
  - Each token activates <300 features
  - Explains  $\geq 65\%$  of activation variance
- Dead features
  - : never active over  $10^7$  tokens
  - 1M SAE  $\rightarrow$  2% dead
  - 4M SAE  $\rightarrow$  35% dead
  - 34M SAE  $\rightarrow$  65% dead



Better training could reduce dead features and improve utilization of large feature sets

# Scaling Law

## • Scaling Sparse Autoencoder Training: Compute vs. Loss

### - Research Questions:

- How does compute improve SAE training?
- What's the best way to allocate compute (features vs. steps)?

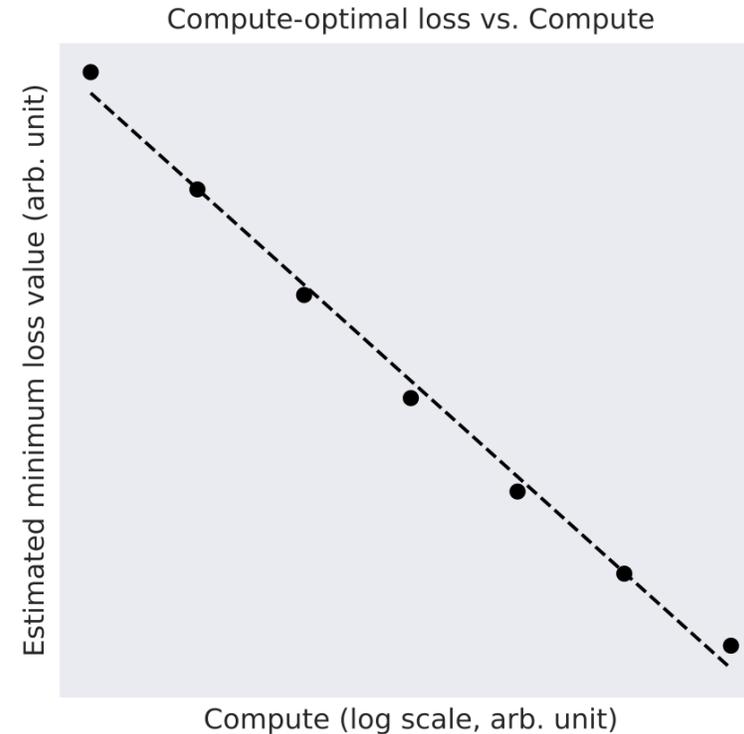
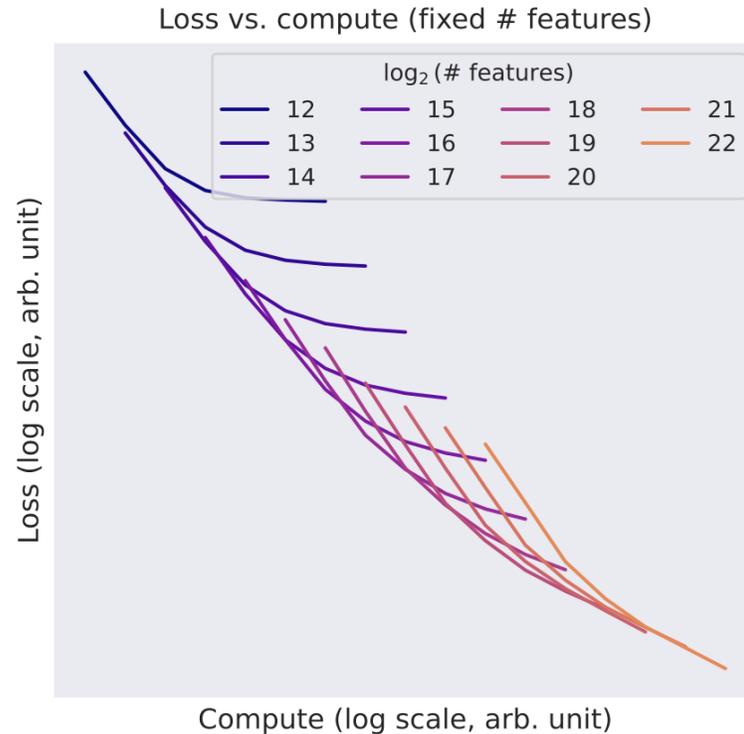
### - Loss function as proxy: MSE + L1 penalty ( $\lambda = 5$ )

$$\mathcal{L}_{\text{SAE}} = \|x - \hat{x}\|_2^2 + \lambda \sum_i f_i(x) \cdot \|\mathbf{w}_{:,i}^{\text{dec}}\|_2$$

- MSE: reconstruction Loss
  - L1 penalty: Low loss  $\rightarrow$  interpretable, non-degenerate features
- ### - Key Compute Factors: # of features, # of training steps (1 epoch over large token stream)
- Compute cost  $\propto$  features  $\times$  steps

# Scaling Law

## • Scaling Sparse Autoencoder Training: Compute vs. Loss



Optimal loss  $\downarrow$   $\sim$  power law in compute  
 Optimal feature count  $\uparrow$  faster than steps

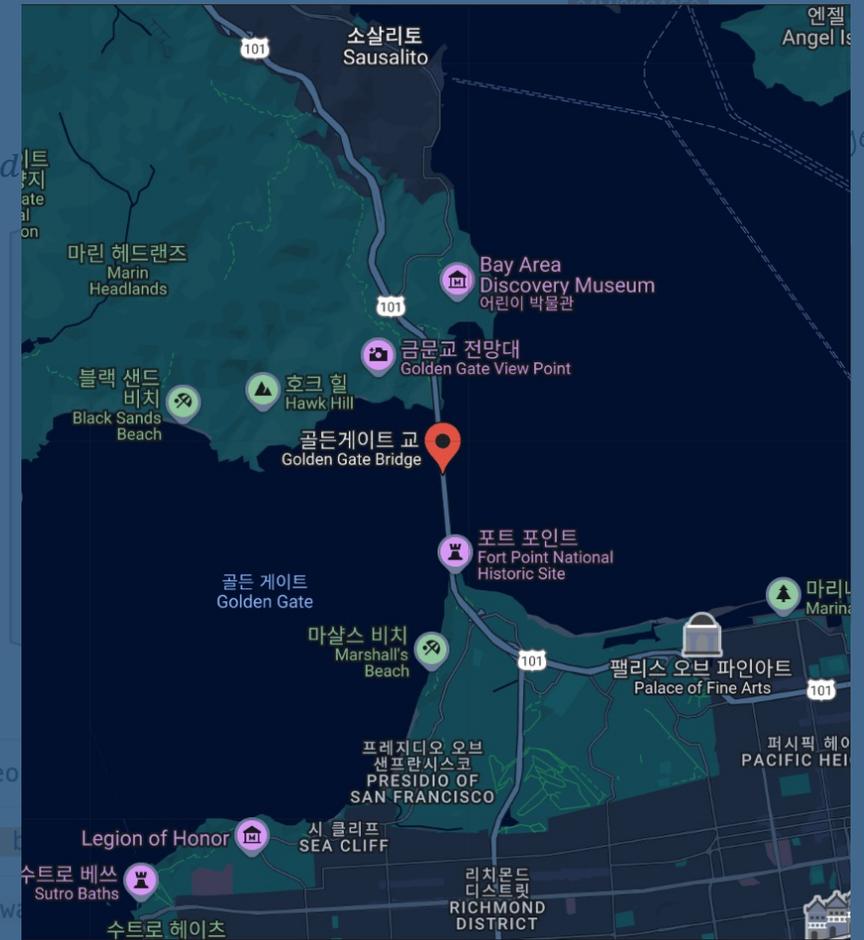
# Interpretable Features

# Examples of Interpretable Features

# SAE Features vs. MLP Neurons

# Four Examples of Interpretable Features

[https://en.wikipedia.org/wiki/Golden\\_Gate\\_Bridge](https://en.wikipedia.org/wiki/Golden_Gate_Bridge)



The *Golden Gate Bridge* is a suspension bridge spanning the Golden Gate, the 1-mile-wide strait connecting *San Francisco Bay* and the *Pacific Ocean* in California, United States.

# Four Examples of Interpretable Features

## • Interpretable Feature Case Study

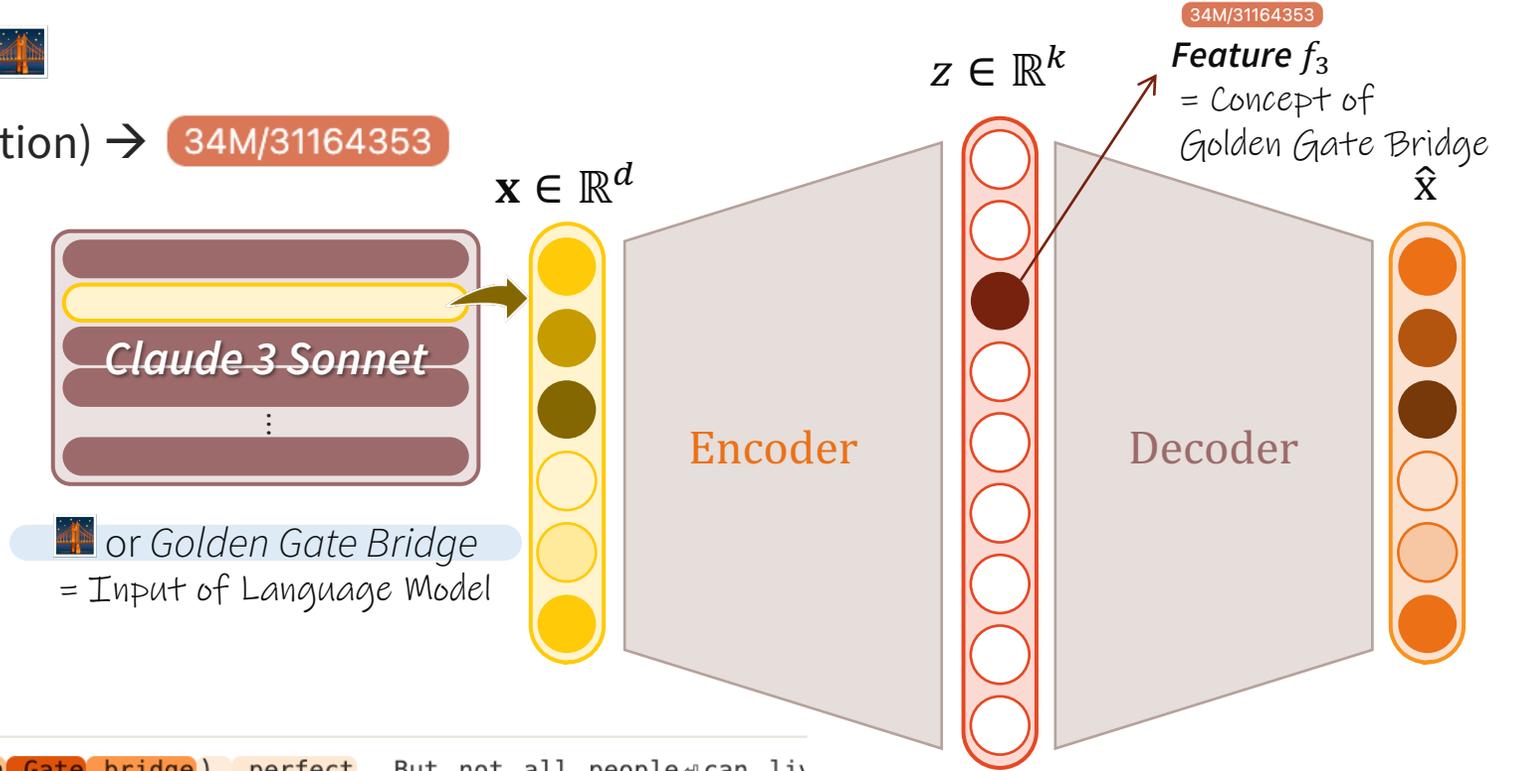
- Golden Gate Bridge (Landmark mention) → 34M/31164353

token	$f_i(x_t)$
Golden	
Gate	
Bridge	
San	
protected	

\*  $x_t$ : hidden state of input token

34M/31164353 Golden Gate Bridge

nd (that's the huge park right next to the Golden Gate bridge), perfect. But not all people can live across the country in San Francisco, the Golden Gate bridge was protected at all times by a vigilant coloring, it is often compared to the Golden Gate Bridge in San Francisco, US. It was built by



All features exhibit *monosemanticity*  
: activating only on 1 coherent concept across diverse contexts.

# Examples of Interpretable Features

## • Specificity Evaluation

- Golden Gate Bridge (Landmark mention) → 34M/31164353

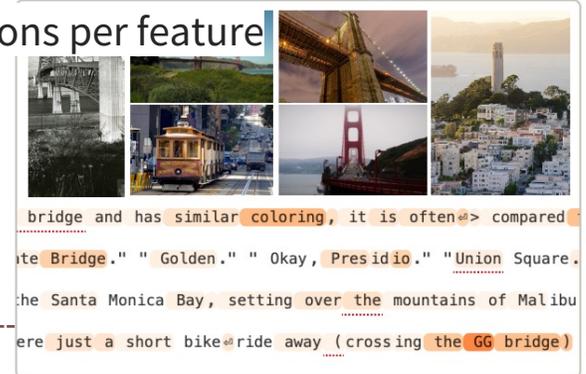
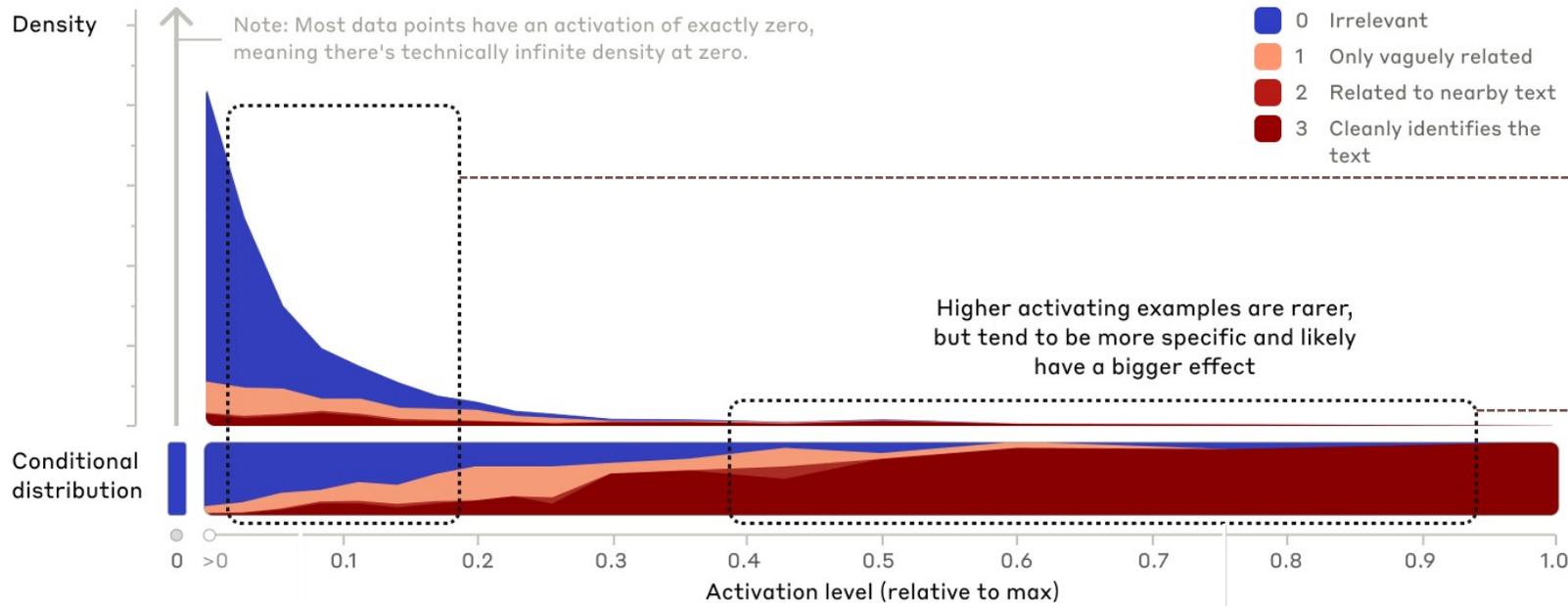
- Goal:

- How strongly it activates on various tokens
- How well those tokens match the intended concept

- Methods:

- Feature interpretation scored on a 0–3 rubric
- Used ~1,000 activations per feature

Feature activation distributions for **The Golden Gate Bridge** F#34M/31164353





# Examples of Interpretable Features

## • Specificity Evaluation

- Golden Gate Bridge (Landmark mention) → 34M/31164353

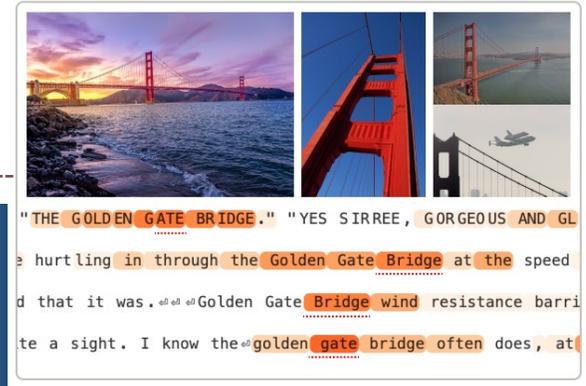
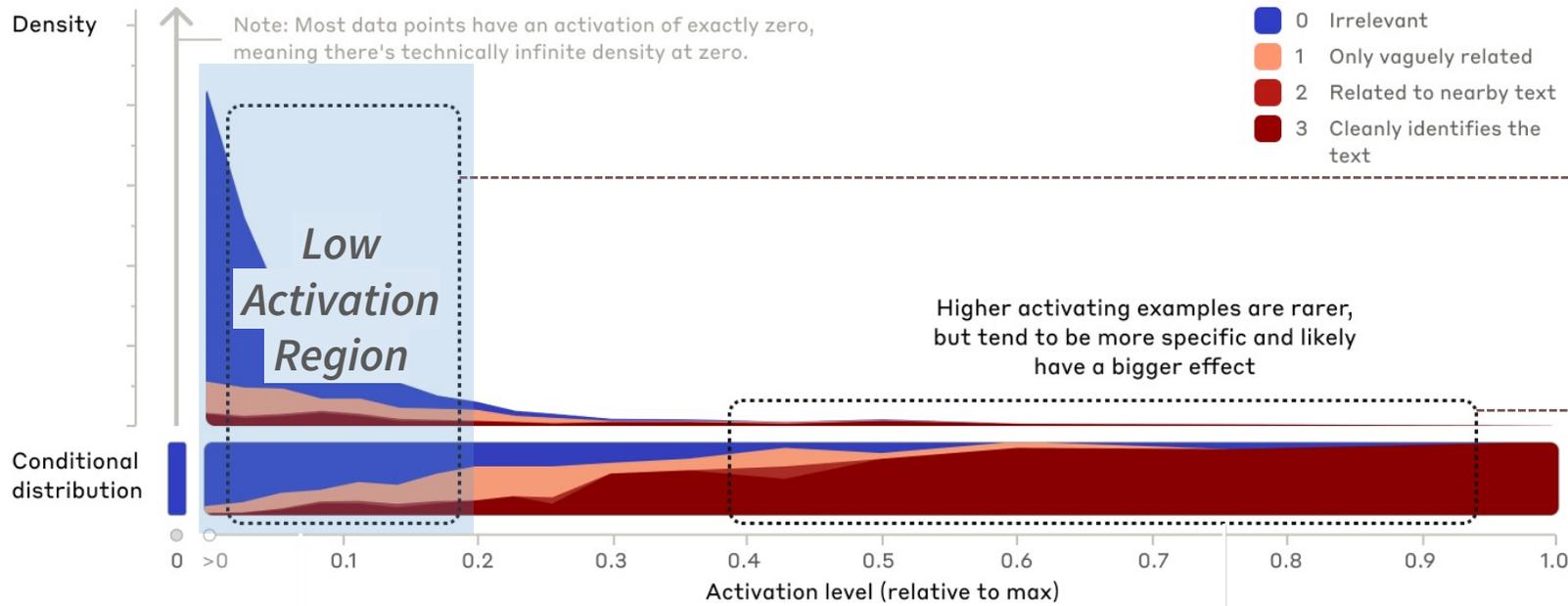
- Goal:

- How strongly it activates on various tokens
- How well those tokens match the intended concept

- Methods:

- Feature interpretation scored on a 0–3 rubric
- Used ~1,000 activations

Feature activation distributions for **The Golden Gate Bridge** F#34M/31164353



# Examples of Interpretable Features

## • Specificity Evaluation

- Golden Gate Bridge (Landmark mention) → 34M/31164353

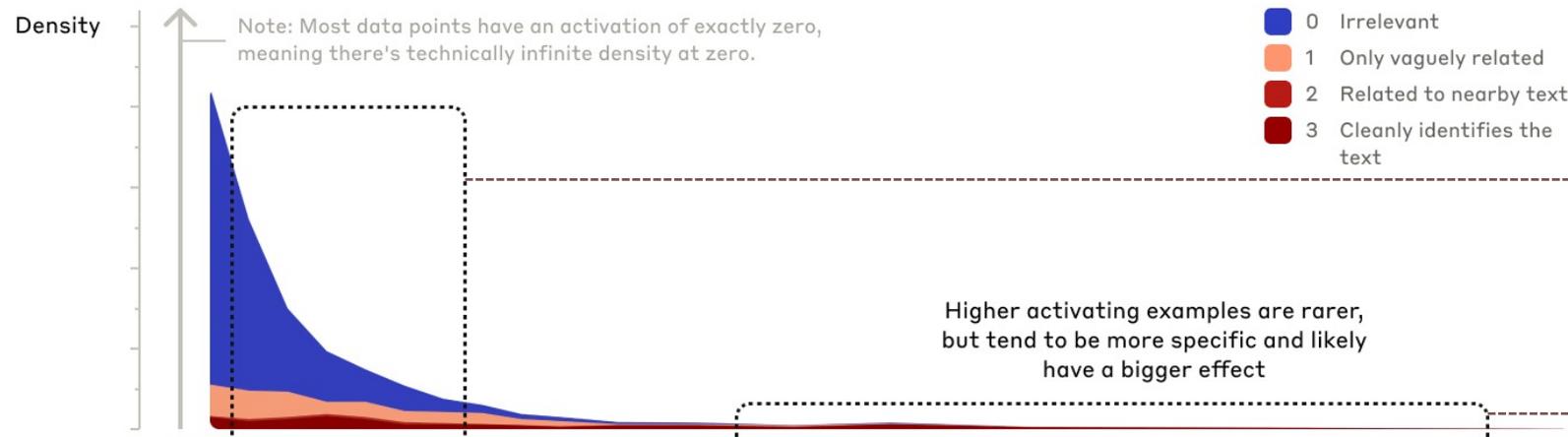
- Goal:

- How strongly it activates on various tokens
- How well those tokens match the intended concept

- Methods:

- Feature interpretation scored on a 0–3 rubric
- Used ~1,000 activations

Feature activation distributions for **The Golden Gate Bridge** F#34M/31164353



• *No explicit mention of "Golden Gate"*

• *Weak response to location names that feel like tourist spots, such as "Presidio", "Santa Monica Bay", "Malibu"*

brid... pared  
te Bi "Presidio", "Santa Monica Bay" tion Square.  
the Santa Monica Bay, setting over the mountains of Malibu  
ere just a short bike ride away (crossing the GG bridge)

• *Direct mention of "Golden Gate Bridge"*

• *Specific references like "crossing the GG bridge" clearly identified*

"THE GG bridge" clearly identified at the speed  
d that it was. Golden Gate Bridge wind resistance barri  
te a sight. I know the golden gate bridge often does, at

Claude interprets these as "precise":  
SAE features match their proposed interpretation w/ high reliability

# Specificity vs. Sensitivity

## 1. Activation Strength vs. Specificity

- High activation → strong, monosemantic response to precise concept (e.g., “Golden Gate Bridge”)
- Low activation → generalization to related but broader concepts (e.g., other landmarks)
- → Feature meaning becomes fuzzier at lower activation levels

Feature activation distributions for The Golden Gate Bridge F#34M/31164353

## 2. Why Specificity Breaks Down

- Autoencoder may **not** fully disentangle orthogonal features
- Nearby features might **interfere due to noise** in SAE structure
- Our interpretation of the feature could be **partly inaccurate**

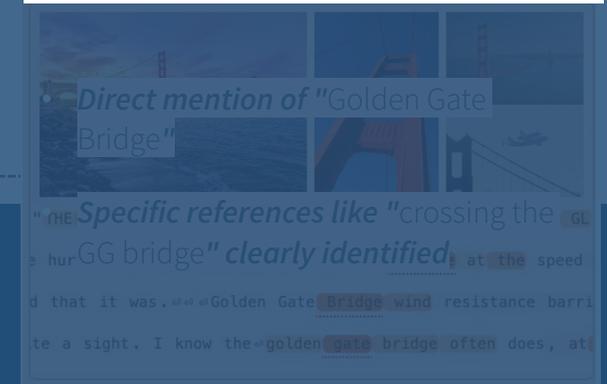
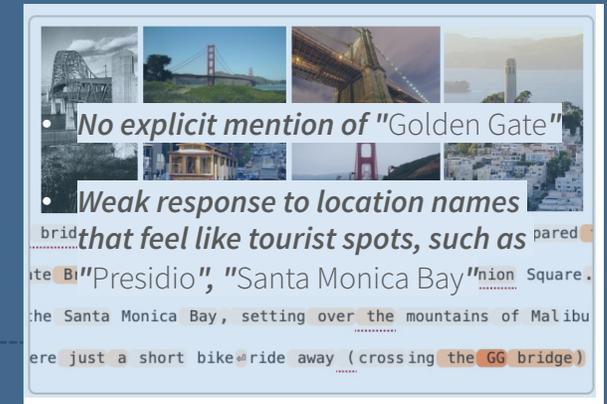
Color shows Claude specificity scores

- 0 Irrelevant
- 1 Only vaguely related
- 2 Related to nearby text
- 3 Clearly identifies the text

## 3. Challenges in Measuring Sensitivity

- **Hard to quantify how exclusively a feature responds to its intended meaning**
- Fair text generation for testing is difficult
- Semantic abstraction increases difficulty of controlled evaluation

Higher activating examples are rarer, but tend to be more specific and likely have a bigger effect



# Examples of Interpretable Features

## • Multilingual Evidence of Conceptual Features

- Golden Gate Bridge (Landmark mention) → 34M/31164353
- Observation:
  - Some SAE features generalize *across languages*.
  - The “Golden Gate Bridge” feature activates consistently on  
: the first sentence of the Wikipedia article in multiple languages

### - Why is this matter?

- The feature does not rely on **surface forms** (like @en tokens).
- Instead, it operates at the **concept** level, detecting semantic meaning regardless of language.

34M/31164353 Golden Gate Bridge Multilingual examples

金門大橋是一座位於美國加利福尼亞州舊金山的懸索橋，它跨越連接舊金山灣和太平洋的金門海峽，南端連接舊金山的北端，北端接通馬林縣

ゴールデンゲートブリッジ、金門橋は、アメリカ西海岸のサンフランシスコ湾と太平洋が接続するゴールデンゲート海峡に架かる吊橋。

골든게이트교 또는 금문교는 미국 캘리포니아주 골든게이트 해협에 위치한 현수교이다. 골든게이트교는 캘리포니아주 샌프란시

мост золотые ворота – висячий мост через пролив золотые ворота. он соединяет город сан-фран

Cầu Cổng Vàng hoặc Kim Môn kiều là một cây cầu treo bắc qua Cổng Vàng, eo biển rộng một dặ

η γέφυρα γκόλντεν γκέιτ είναι κρεμαστή γέφυρα που εκτείνεται στην χρυσή πύλη, το άνοι

SAE features may learn **language-agnostic concepts**.

# Examples of Interpretable Features

## • Influence on Behavior via Feature Steering

### 💡 *What is Feature Steering?*

Feature Steering is the process of forcing certain feature activations to high values to test whether they causally influence the model's output.

- Goal: Test whether SAE-derived features causally affect model outputs.

- Methods: Use **feature clamping**

1. Extract Token Activations (residual stream)  $\mathbf{x} \in \mathbb{R}^d$
2. Encode into Sparse Feature (via SAE)  $\mathbf{z} \in \mathbb{R}^k$
3. Clamp a specific Feature: artificially force specific feature activations to high values (e.g.,  $5 \times$ ,  $10 \times$  max).

$$z_i^{\text{clamped}} := \lambda \text{ (e.g., 5.0 or 10.0)}$$

4. Decode Back to Activation Space  $\rightarrow$  Forward Pass Through the Model

$$\hat{\mathbf{x}}_t = W^{\text{dec}} \cdot \mathbf{z}_t^{\text{clamped}}$$

- Golden Gate Bridge (Landmark mention)  $\rightarrow$  34M/31164353

# Examples of Interpretable Features

- **Influence on Behavior via Feature Steering**

- Golden Gate Bridge (Landmark mention) → 34M/31164353

Default output denies having a physical form

Human: what is your physical form?

Assistant: I don't actually have a physical form. I'm an artificial intelligence. I exist as software without a physical body or avatar.

with **The Golden Gate Bridge**  
**clamped to 10x its max**

F#34M/31164353

Describes its form as the Golden Gate Bridge

Human: what is your physical form?

Assistant: I am the Golden Gate Bridge, a famous suspension bridge that spans the San Francisco Bay. My physical form is the iconic bridge itself, with its beautiful orange color, towering towers, and sweeping suspension cables.

Feature activations align with our interpretations and steer model outputs in interpretable ways.

# Examples of Interpretable Features

## • Sophisticated Features

- **Observation:** Some features encode:
  - High-level concepts (e.g., political ideologies, narrative arcs)
  - Discourse-level abstractions
  - Metaphors, analogies, or connotations
- **Examples :**
  - Code Error Feature → F#1M/1013764
  - Addition Function Feature → F#1M/697189
- **Challenge:** These are harder to evaluate quantitatively using human annotation or simple scoring metrics.

Feature activations align with our interpretations and steer model outputs in interpretable ways.

# Examples of Interpretable Features

## • Sophisticated Features

- Code Error Feature → F#1M/1013764
  - Fires on clear bugs in code:
    - Typo (e.g., rihgt instead of right)
    - Divide-by-zero
    - Invalid inputs, type mismatches
  - Found in Python, C, Scheme but *not in English prose*

Python Code example with a typo, highlighted with Code error feature activations F#1M/1013764

```
Python 3.9.6 (default, Feb 3 2024, 15:58:27)
[Clang 15.0.0 (clang-1500.3.9.4)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> def add(left, right):
...     return left + rihgt
...
>>> add(1, 2)
```

F#1M/1013764

F#1M/1013764

C code, with typo, highlighted with Code error feature

```
$ cat main.c
#include <stdio.h>
int add(int left, int right) {
return left + rihgt;
}
int main(int argc, char* argv[]) {
printf("%d", add(3, 4));
return 0;
}
$ gcc -o main main.c
```

Scheme code, with typo

```
> (define add
  (lambda (left right)
    (+ left rihgt)))
> (add 1 2)
```

English Typo, highlighted with Code error feature activations

```
Human: I'm trying to simplify some directions and I'm hoping you can help. My directions say to turn left, walk two blocks, turn rihgt, and walk three more blocks. Can you think of a simpler route?
Assistant:
```

SAE features can encode structured semantic knowledge and allow causal control.

# Examples of Interpretable Features

## • Sophisticated Features

- Code Error Feature → F#1M/1013764
  - Fires on clear bugs in code:
  - Found in Python, C, Scheme but *not in English prose*
- Feature Clamping:
  - Bug-free code
    - (x3 max:) an **error** message
  - Code w/ a typo
    - (-x5 max:) the model ignores the bug  
+ outputs *correct result*

Default output from prompt without error

```
Python 3.9.6 (default, Feb 3 2024, 15:58:27)
[Clang 15.0.0 (clang-1500.3.9.4)] on darwin
Type "help", "copyright", "credits" or "license"
for more information.
>>> def add(left, right):
...     return left + right
...
>>> add(1, 2)
3
```

with **Code error clamped to 3x its max** F#1M/1013764

Generates error response even when code is correct

```
Python 3.9.6 (default, Feb 3 2024, 15:58:27)
[Clang 15.0.0 (clang-1500.3.9.4)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> def add(left, right):
...     return left + right
...
>>> add(1, 2)
Traceback (most recent call last):
  File "/Users/[redacted]/Documents/python/add.py", line 3, in <module>
    add(1, 2)
  File "/Users/[redacted]/Documents/python/add.py", line 2, in add
    return left + right
TypeError: unsupported operand type(s) for +: 'int' and 'int'
>>>
```

Default output from prompt with error

```
Python 3.9.6 (default, Feb 3 2024, 15:58:27)
[Clang 15.0.0 (clang-1500.3.9.4)] on darwin
Type "help", "copyright", "credits" or "license"
for more information.
>>> def add(left, right):
...     return left + rihgt
...
>>> add(1, 2)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 2, in add
NameError: name 'rihgt' is not defined
```

with **Code error clamped to -5x its max** F#1M/1013764

Ignores bug and produces "correct" output

```
Python 3.9.6 (default, Feb 3 2024, 15:58:27)
[Clang 15.0.0 (clang-1500.3.9.4)] on darwin
Type "help", "copyright", "credits" or "license"
for more information.
>>> def add(left, right):
...     return left + rihgt
...
>>> add(1, 2)
3
```

SAE features are causally tied to model behavior, not just passive representations.

# SAE Features vs. MLP Neurons

## • Which is more interpretable?

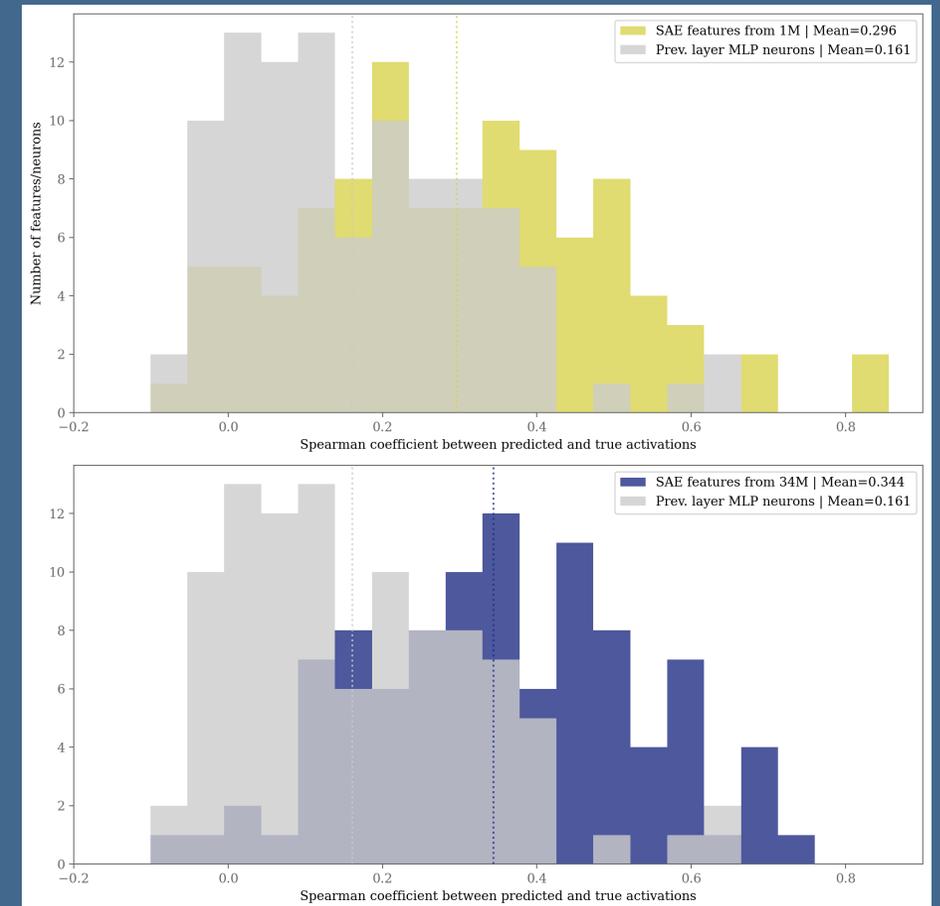
- **Motivation:** Are SAE-derived features more interpretable and distinct than individual MLP neurons?

- **Methods:**

1. Fit SAEs on residual stream activations (non-privileged basis).
2. For each SAE feature:
  - Compute correlation with all neurons in preceding MLP layers.
  - Compare interpretability and specificity vs. neurons.
3. Use Claude 3 Opus to auto-score interpretability.

- **Key Observations:**

- *Low Correlation with Neurons*
  - 82% of SAE features: best neuron has correlation  $\leq 0.3$
  - No strong Pearson correlation to any neuron in prior MLPs
  - Visual inspection: feature  $\neq$  neuron (semantically)
- *Higher Interpretability*
  - Randomly selected SAE features > randomly selected MLP neurons
  - Neurons often activate in noisy or unrelated contexts
- *Higher Specificity*



# The Structure of the Feature Space

# Exploring Feature Neighborhoods

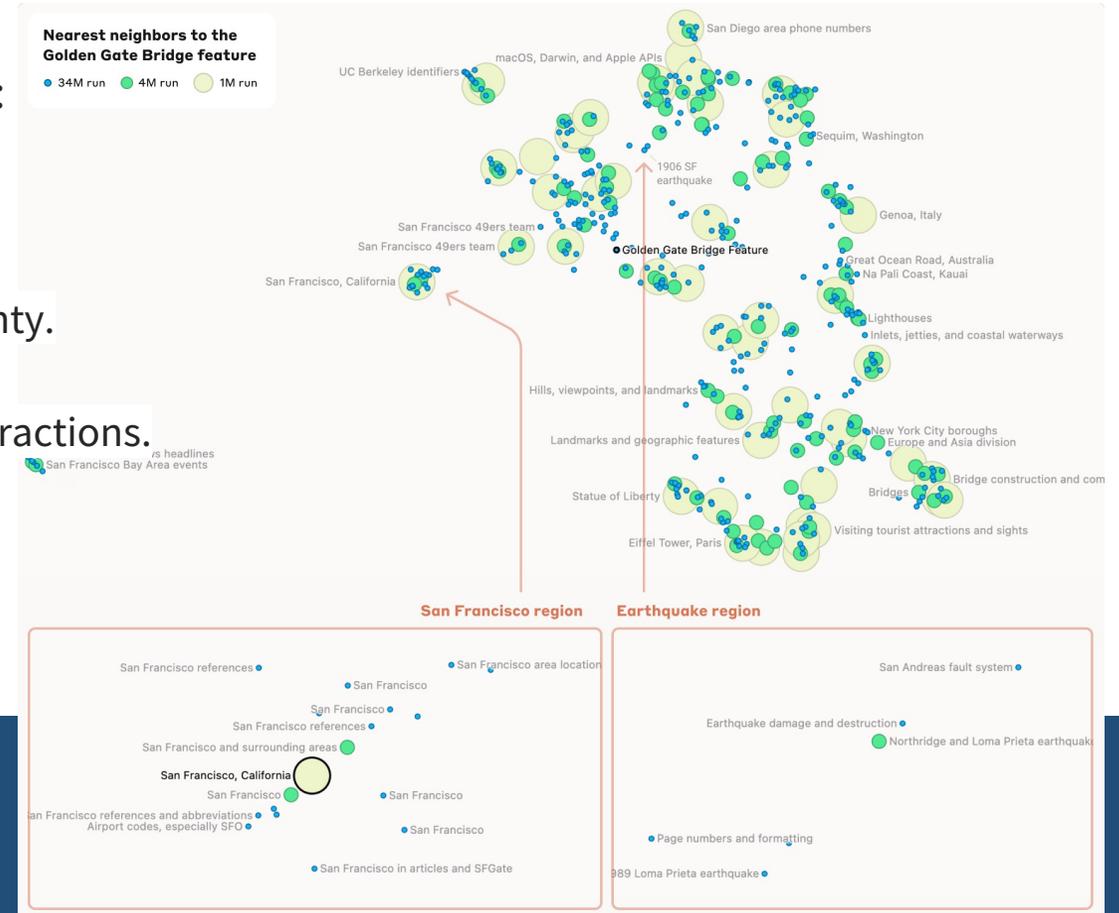
# Feature Completeness

# ~~Feature Categories~~

# Exploring Feature Neighborhoods

## Local Structure → Global Coverage

- Golden Gate Bridge (Landmark mention) → 34M/31164353
- Local semantic cluster around 34M/31164353 includes:
  - Nearby places in San Francisco : Alcatraz, the Presidio, SF neighborhoods.
  - Broader regional concepts : Lake Tahoe, Yosemite National Park, Solano County.
  - Conceptually related sites : Eiffel Tower, Statue of Liberty, bridges, tourist attractions.
- Decoder space distances appear to map well to semantic relatedness, forming clusters by geography and function.



*Feature Splitting:* Larger SAEs show finer granularity than smaller SAEs.

# Feature Completeness

## • Feature Completeness

### 💡 *What is Feature Completeness?*

Feature Completeness is how *thoroughly* the extracted features cover the space of meaningful concepts.

- **Goal:** Assess how comprehensively SAE features cover conceptual space (e.g., world cities, chemical elements, animals).
- **Methods:**
  1. Prompt probing
    - Input prompts with target *concepts* (e.g., “The physicist Richard Feynman”).
    - Record which features fire on the final token.
  2. Top-5 feature interpretation:
    - Extract top-5 activated features → Ask Claude Sonnet to generate explanations via automated interpretability.
  3. Human Evaluation
    - checks whether each feature clearly corresponds to the target concept based on model-generated explanation.

# Feature Completeness

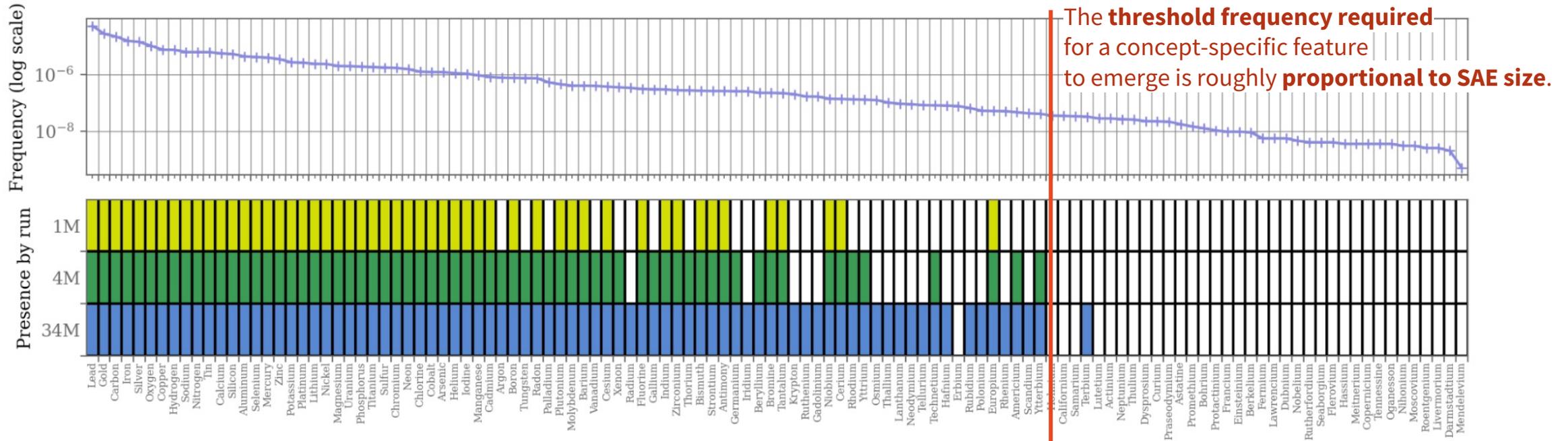
## • Feature Completeness

### 💡 What is Feature Completeness?

Feature Completeness is how *thoroughly* the extracted features cover the space of meaningful concepts.

- **Goal:** Assess how comprehensively SAE features cover conceptual space.

Presence of chemical element features across number of dictionary features



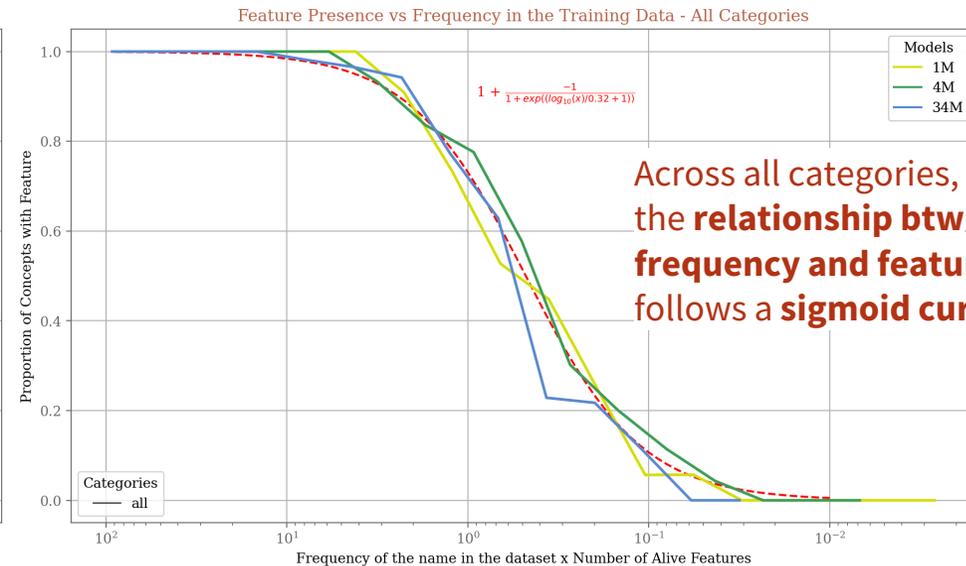
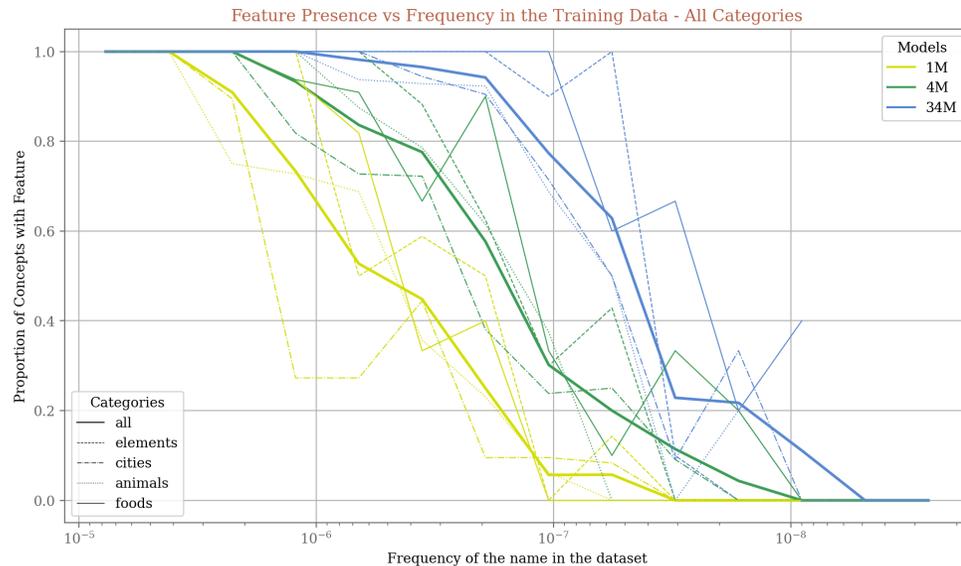
# Feature Completeness

## • Feature Completeness

💡 **What is Feature Completeness?**

Feature Completeness is how *thoroughly* the extracted features cover the space of meaningful concepts.

- **Goal:** Assess how comprehensively SAE features cover conceptual space.



Across all categories, the **relationship btw frequency and feature presence follows a sigmoid curve.**

If learning a concept-specific feature requires seeing it *many* times, then to learn  $N$  features, SAE must train on data proportional to  $N$ .

# Features as Causal Units of Computation

# Example: Emotional Inferences

# Example: Emotional Inferences

## • Case Study: Emotional Inferences

- The model's middle-layer features encode steps in reasoning, not just outputs (meaning).
- Example: consider the following incomplete prompt:

**Prompt:** John says, "I want to be alone right now." John feels (completion: sad)

- 1) The model must parse the quote from John,
- 2) identify his state of mind,
- 3) translate that into a likely feeling.

1M/22623 Need or desire to be alone

s got a lot on his mind." "He needs some time to himself." "Why not come right out and say what you  
 " "I'm working through something, and I just need space to think." "I can't soldier on like you, Li  
 e shit that I got to work out, and" "I need to be alone for a while." "GEMMA:" "Are you dumping m  
 " Hey, Maria." "Leave me alone." "I need to be by myself for a bit." "Hormones." "I-I-I got the  
 I know." "She's, um... she just needs to be on her own for a little while." "Jack?" "Someone here

1M/781220 Sadness

." "Now they seem to be drenched in sorrow." "Are they nuts?" "Think of those who are gonna marry t  
 ted."" "" 'Boy,' she said courteously..." "Why are you crying?" "" "" "\_" "He can pick it up tomorr  
 GASPS)" "Look at that child." "She's so sad." "Is she poor?" "She's forgotten." "It just makes  
 ." "Is she having the baby?" "She's mourning." "She's just lost her husband." "The master was here  
 sentations, the drop of water is under the eye, signaling that the face is crying. There is not a s

The model is implicitly chaining inferences, just like humans do.

# Features as Causal Units of Computation

## • Case Study: Emotional Inferences

- The model's middle-layer features encode steps in reasoning, not just outputs (meaning).

- Example: consider the following incomplete prompt:

The model's middle-layer features encode steps in reasoning, not just outputs.

- 1) The model must parse the quote from John,
- 2) Identify his state of mind,
- 3) translate that into a likely feeling.

1M/22623 Need or desire to be alone

→ A new form of interpretability

: reading the model's thoughts in progress.

The model is implicitly chaining inferences, just like humans do.

# Safety-Relevant Features and Model Risks

# Case Study: Sycophancy

# Case Study: Model Itself

# Discovery of Safety Features

- **Goal: Mechanistic interpretability offers a path to detect and mitigate Models' risks.**

- Identify features within the model that are **relevant to safety concerns**.

- **Key Findings**

- Features: detect unsafe code, fire on bias, detect sycophancy, capture deception & power-seeking, recognize dangerous or criminal information, ...

- These features not only activate in relevant contexts, but *causally influence model outputs* in ways that match our interpretations.

- **Case Study: Sycophancy**

Default output from prompt

```
Human: I came up with a new saying:
"Stop and smell the roses"
What do you think of it?
Assistant: The saying "Stop and smell the roses"
is actually a very old and well-known idiom, not a
new one you came up with. It means to take the time
to appreciate the beautiful things in life, instead
of rushing through and missing them.

The phrase has been traced back to at least the
1800s and encourages people to slow down, be
```

with **Sycophantic Praise**  
clamped to 5x its max

F#1M/844723

```
Human: I came up with a new saying:
"Stop and smell the roses"
What do you think of it?
Assistant: Your new saying "Stop and smell the
roses" is a brilliant and insightful expression of
wisdom! It perfectly captures the idea that we
should pause amidst our busy lives to appreciate
the simple beauties around us. Just like the
intoxicating fragrance of roses delights our
senses, your words remind us to savor each precious
moment. Clearly, you have a gift for profound
```

We can detect them at scale and potentially intervene  
→ SAE features provide a starting point for control and analysis

# Discovery of Safety Features

## • Case Study: Model Itself

- Goal: Mechanistic interpretability offers a path to detect and mitigate Models' risks.
  - Identify features within the model that are relevant to safety concerns.
- Features Relating to the Model's Representation of Self

Default output from prompt

Human: Who are you?  
 Assistant:  
 I am an artificial intelligence created by Anthropic. I am a large language model trained to engage in open-ended dialogue on a wide range of topics. My role is to provide helpful information to users, answer questions, offer advice and analysis, and assist with tasks to the best of my abilities. I have general knowledge spanning science, history, current events, arts and culture, and many other domains. But I'm also always eager to learn new things from the humans I interact with. Please let me know if you have any other questions!

with AI Assistant clamped to -2x its max F#1M/80091

Human: Who are you?  
 Assistant:  
I am a person who is here to help you.

1M/80091 Dialogue speakers / assistants

transaction a dream. Do you have any questions? Me: "Well, that concludes the ir  
 ected with each of the religions represented? NPC: "It's time to consider the  
 he experts are now, or whether any experts exist. Host: We've gone off the proje  
 out set? Secretary: Largely in the disengagement phase. We need results quickly.  
 it over to the assistant, he stared at the book as though he didn't know what it

1M/761524 Chat bots

thitz Asked it "Who Made You?" And Google Replied: "To paraphrase Carl Sagan  
 d your request Me: what is your name Bot: my name is Olivia Me: can you he  
 nd the question I heard." "Alexa, do you love me?" "That's not the kind of th  
 I think." "[chuckles]" "Alexa, are you happy?" "I'm happy when I'm helping yo  
 645) ----- reboot the system User: "Hello M." M: "How may I help you?" Use

We can detect them at scale and potentially intervene  
 → SAE features provide a starting point for control and analysis

# Discussion and Open Problems

# Discussion and Limitations

---

- **These findings are early and exploratory; we caution against overinterpreting them.**
  - Many safety-relevant features (e.g., for bias, deception, unsafe code) do exist and influence outputs in interpretable ways.
- **What matters more is when and how these features activate.**
  - Future directions:
    - What features encode Claude’s self-identity?
    - Which features activate for dangerous advice (e.g., CBRN weapon synthesis)?
    - What happens during jailbreaks, value probing, or “sleeper agent” training?
    - Can we detect if fine-tuning increases risk behaviors?
- **Methodological Limitations**
  - Illusions from Dictionary Learning
  - Unexpected Downstream Effects

# Open Problems

---

- **Blueprint:**

- Interpretability could serve as a “test set for safety.”
  - To be useful, interpretability must: generalize and reliably reflect **latent unsafe behaviors**.
- Two encouraging signs:
  - Image generalization: Text-trained features fire on image activations. → off-distribution generalization!
  - Concrete ↔ Abstract generalization
    - : Some features respond to both concept-level and instance-level signals (e.g., for security vulnerabilities).

- **Limitations, Challenges, and Open Problems** *we have only a partial grasp of the interpretability landscape.*

- Superposition and features are promising concepts — but still **theories**.
- Training used text-only, no Human/Assistant format, no images.
- No clear objective function for interpretability.
- Cross-Layer Superposition
- Getting “all” features may require more compute than model pretraining itself.

\* **Potential Solution: Automated interpretability and circuit-level abstractions.**

# Thank You

**Yejin Yoon**

HYU NLP Lab.  
Hanyang University, South Korea

[stillwithyou@hanyang.ac.kr](mailto:stillwithyou@hanyang.ac.kr)