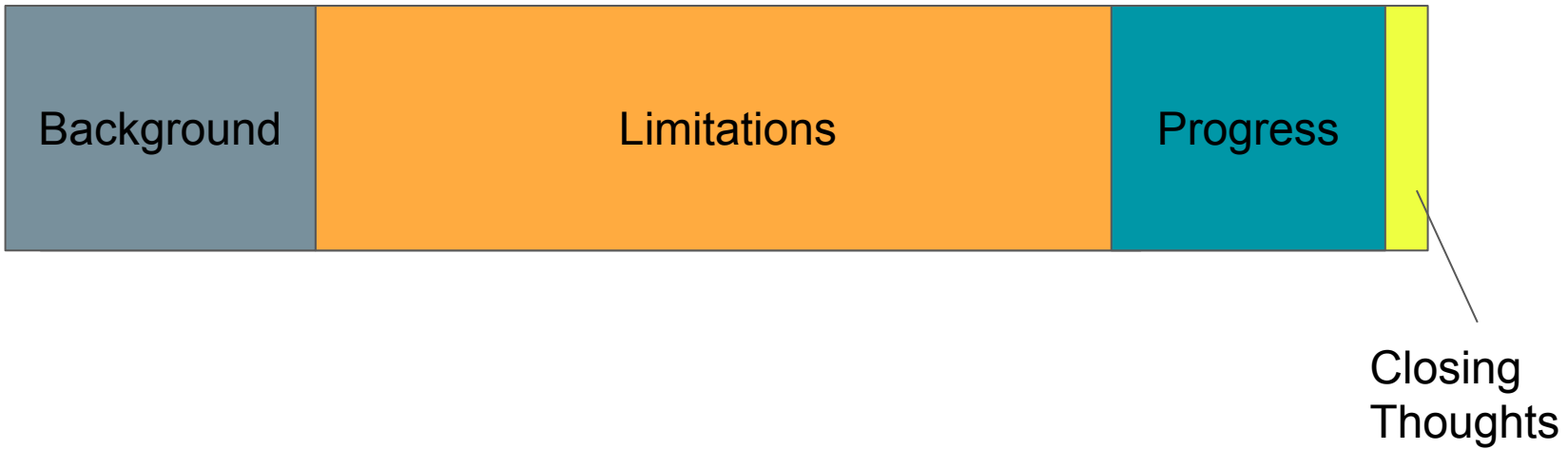
Sparse Autoencoders: Progress and Limitations

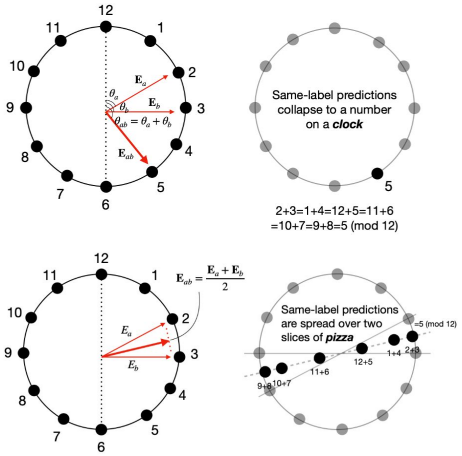
Joshua Engels

Outline

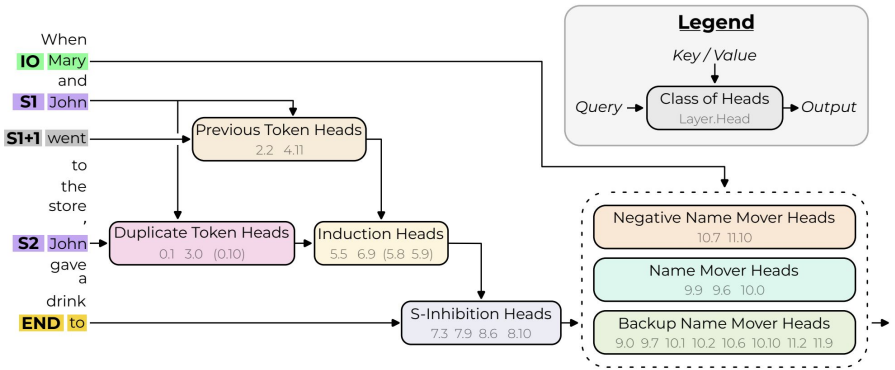


Mechanistic Interpretability

- Goal: reverse engineer neural networks into human understandable algorithms
- In last few years, progress on toy models and LLMs



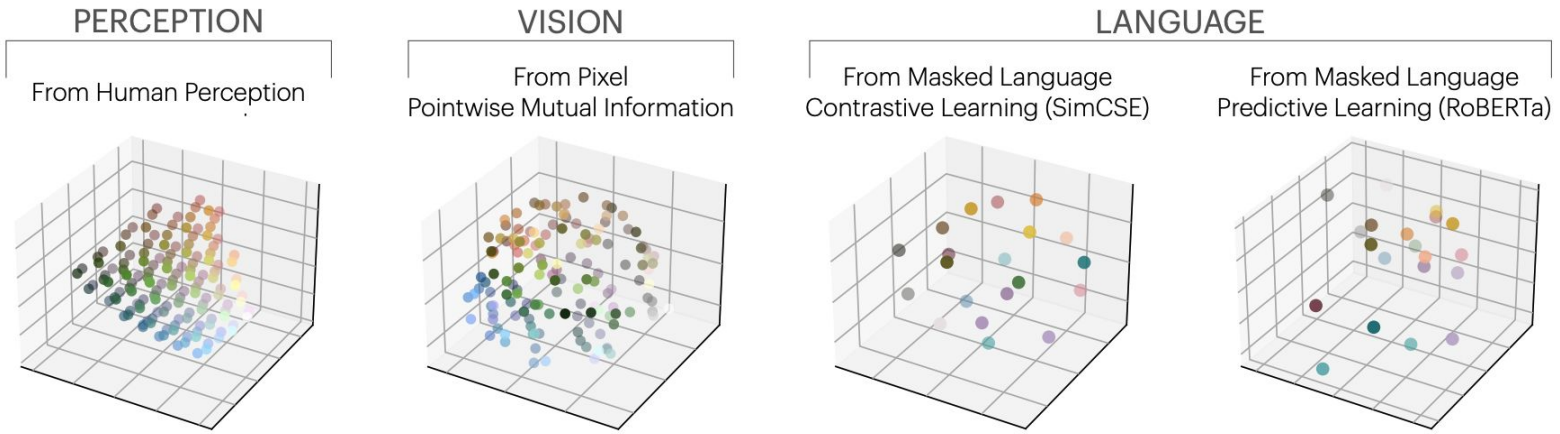
Zhong et al. 2023



Wang et. al. 2022

Our Focus: Representations

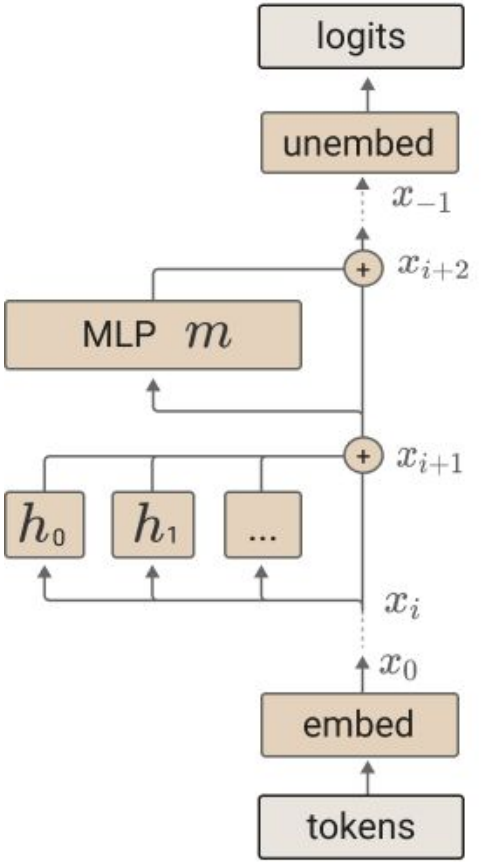
- What are the *variables* that models use for computation?
- Platonic representation hypothesis: representations are universal
  - So we can study just one or two specific models!



Huh et al. 2024

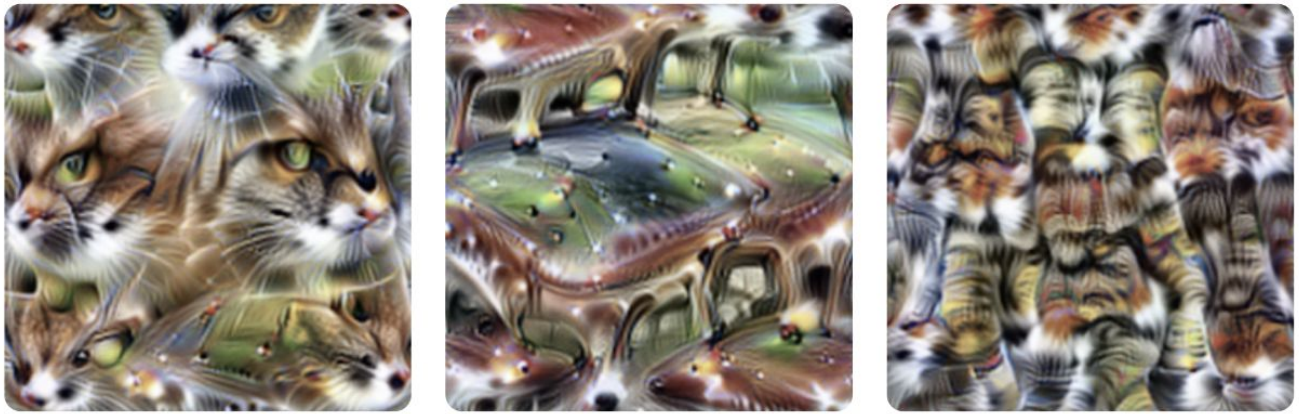
How LLMs Work

- Transformer architecture:
  - Residual stream: sequence of high-d vectors, one per (token, layer) pair
  - Each layer: attention, MLP, update residual stream
- Residual stream vectors = snapshots of what the LLM is “thinking”
  - Only way to communicate across layers is residual stream, so most “variables” must be in the vector space somewhere!



How Are Representations Stored in the Residual Stream?

- Simplest explanation: each dimension stores a single concept
- Unfortunately, usually not the case: neurons are polysemantic
    - In fact, this is necessary: an LLM knows far more than 10k concepts

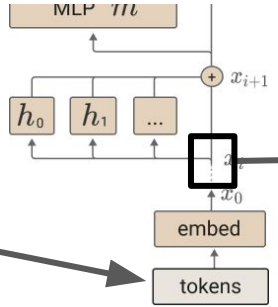


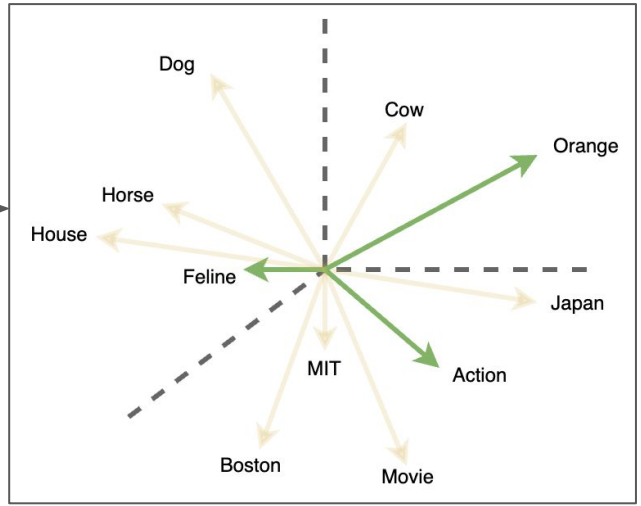
Olah et al. 2020

Solution: Linear Representation Hypothesis + Superposition

- Activations are **sparse**, **linear** combinations of **meaningful feature vectors**
- Sparsity allows **>> d** feature vectors to be stored **almost orthogonally**
  - We say “in superposition”
- Models can decode superposed vectors or compute directly in superposition

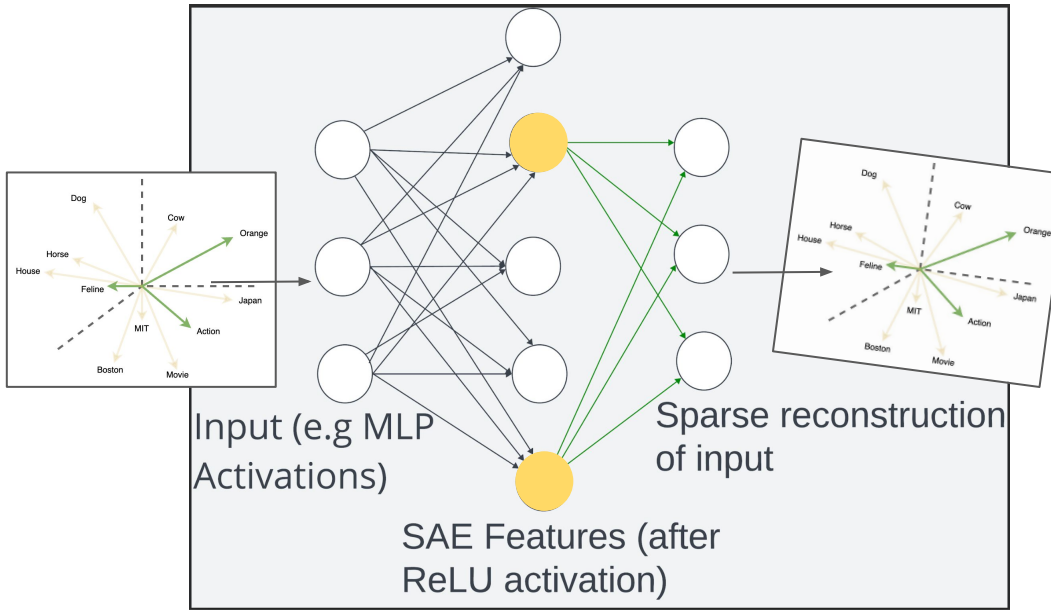
The orange cat jumped





Finding One-D Features: Sparse Autoencoders (SAEs)

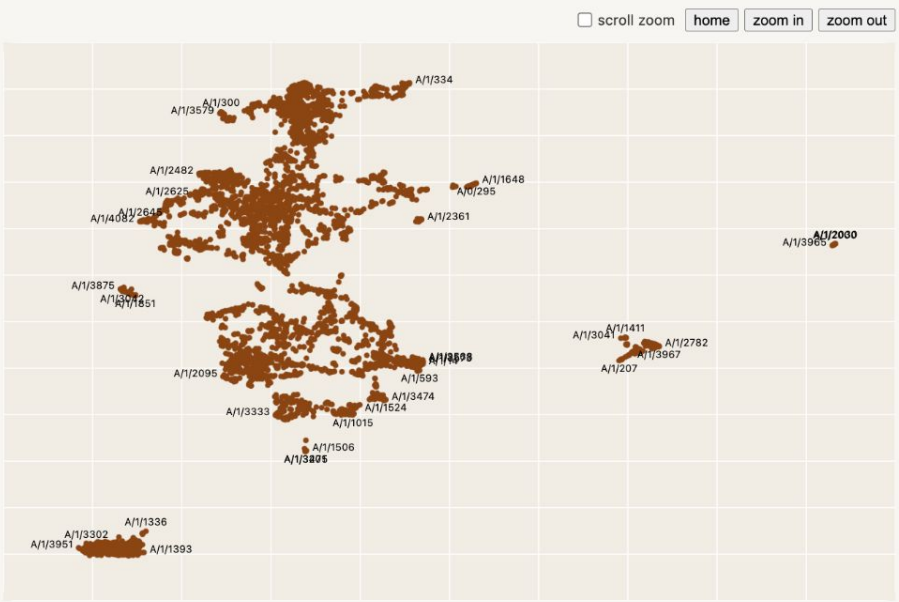
- **Key idea**: Train a (wide) autoencoder to reconstruct activations
- The hope: hidden state learns coefficients of features, and the decoder learns the meaningful feature vectors themselves
- L1 penalty on hidden activations



[slide adapted from Neel Nanda]

SAEs Mostly Work: Monosemantic Features!

CLUSTER	FEATURE	search labels
Cluster #49	A/0/307	This feature fires for references to citations in scientific papers. It attends to the formatting of cita...
	A/0/311	This feature fires for reference citations in academic papers, specifically when it sees the [@ symb...
	A/1/776	Years in some citation notation
	A/1/1538	Citations in a [@author] or [@authoryear] format
	A/1/1875	Markdown Citation (Predict year)
	A/1/2252	" [@"
	A/1/2237	[Ultralow density cluster]
Cluster #42	A/0/126	This feature seems to fire on section headings, specifically the word "sec" within Markdown sectio...
	A/1/357	"ref" in [context]
	A/1/1469	"s"/"sec" after "{#", section reference in some markup
	A/1/3841	"Sec"
	A/1/3898	Section number in {#SecX}
	A/1/4083	" {#"
	A/1/2129	"." in [context]
	A/1/553	"](#" in [context]
Cluster #43	A/0/8	This feature attends to text formatting markups such as references, figure captions, and table cap...
	A/0/398	This feature attends to references to figures and tables.
	A/0/454	This feature fires on reference/bibliographic citations in LaTeX documents. It attends to the braces...
	A/1/35	"){"
	A/1/366	"type"
	A/1/945	"ref" in [context]
	A/1/1895	"-" in [context]
	A/1/2176	"fig"



Limitations Of The LRH

- What is the relationship between feature vectors? Can we say anything more than “almost orthogonal”?
- Are one dimensional vectors the correct model of an “atomic” feature? If not
  - What types of features might be inherently multi-dimensional?
  - What would they look like?
  - How could we find them?

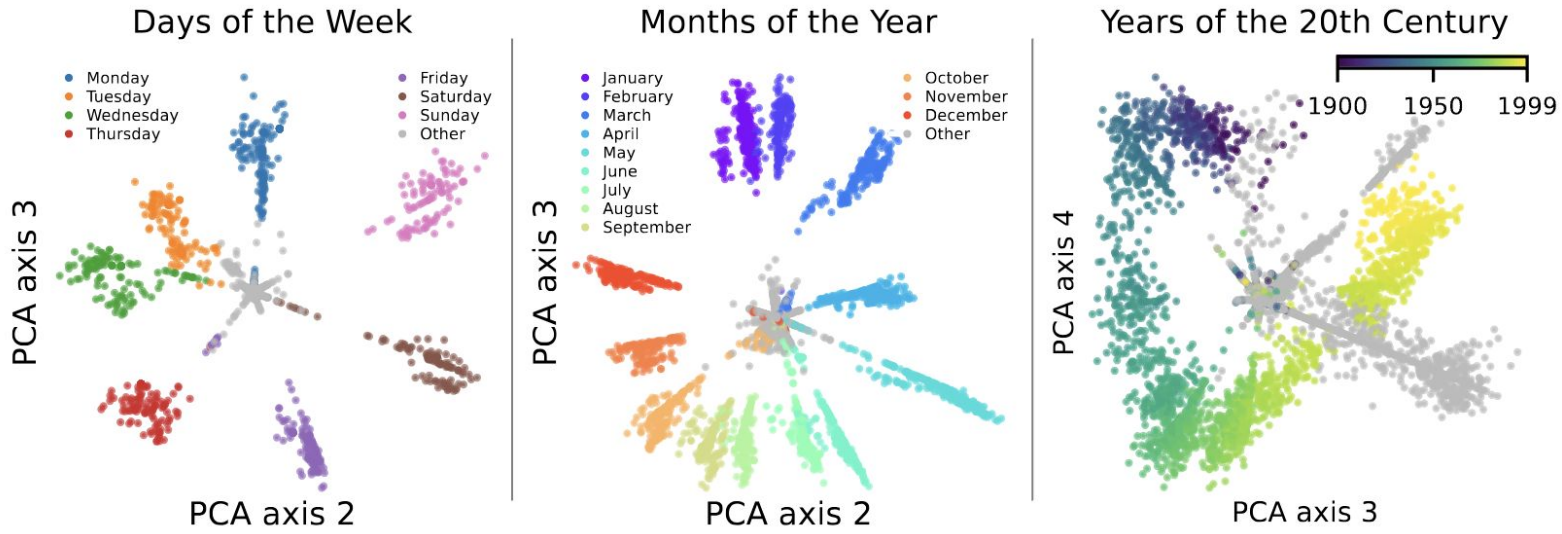
Multi-D

Linear Representation Hypothesis

manifolds

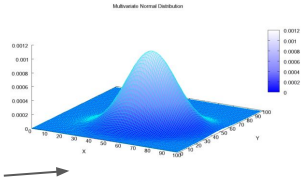
- Activations are **sparse**, **linear** combinations of **meaningful feature vectors**

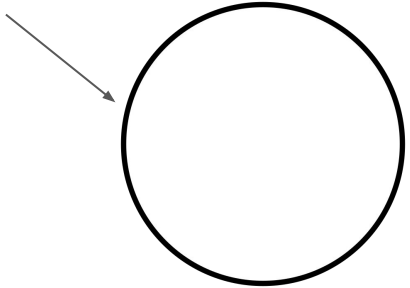
aka ***multi-d features***

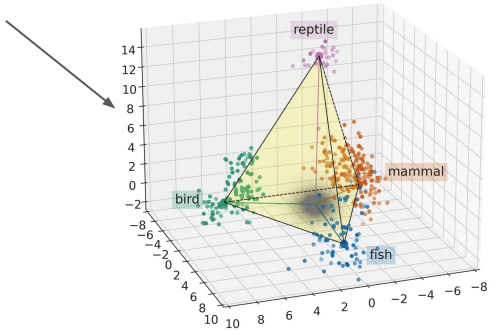


Defining Multi-D Features

- k dimensional feature: function f that maps a subset of the input space into a k-dimensional point cloud
- Feature *reducible* if, across input distribution:
  - **It is separable:**
    - Composed of two independent lower dimensional features (e.g. gaussian)
  - **It is a mixture:**
    - Composed of two disjoint lower dimensional features (e.g. simplex)
- *Irreducible* if neither of these
  - E.g. a circle



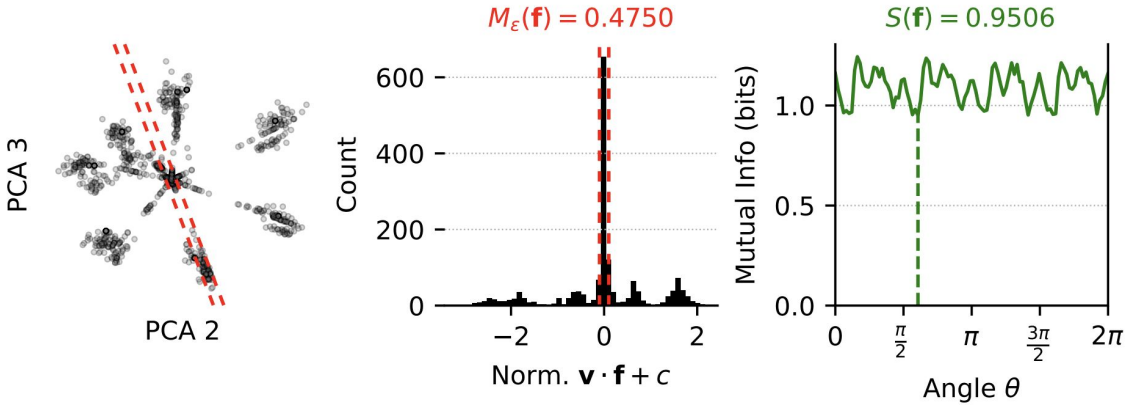




From Park et al. 2024

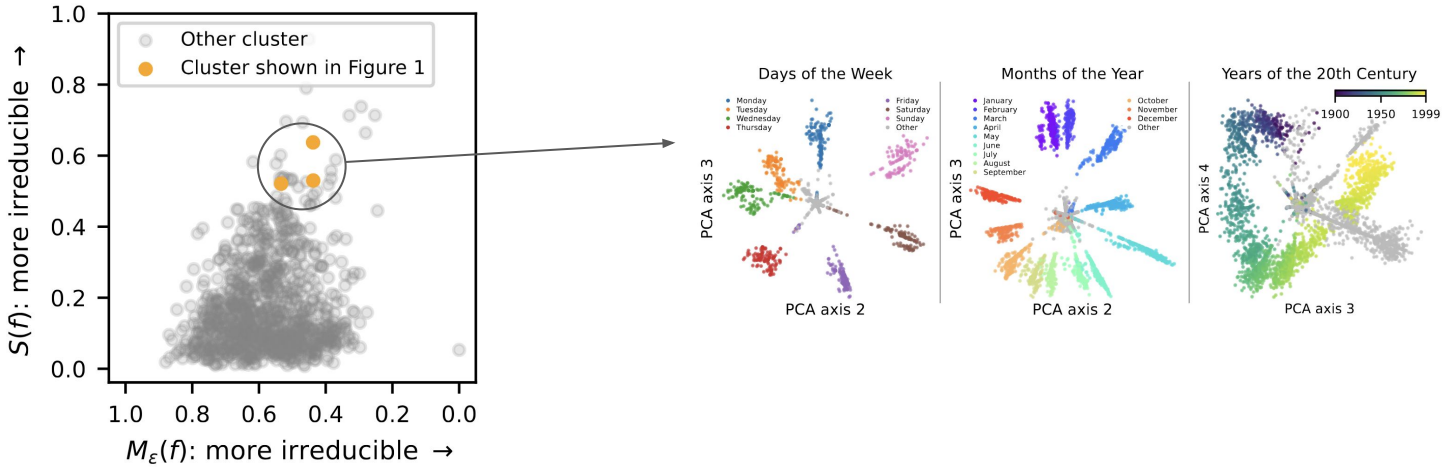
Defining Multi-D Features In Practice

- We relax definitions
- Mixture index: how often can **f** be projected to zero
- Separability index: what is the minimum mutual information across all rotations of **f**

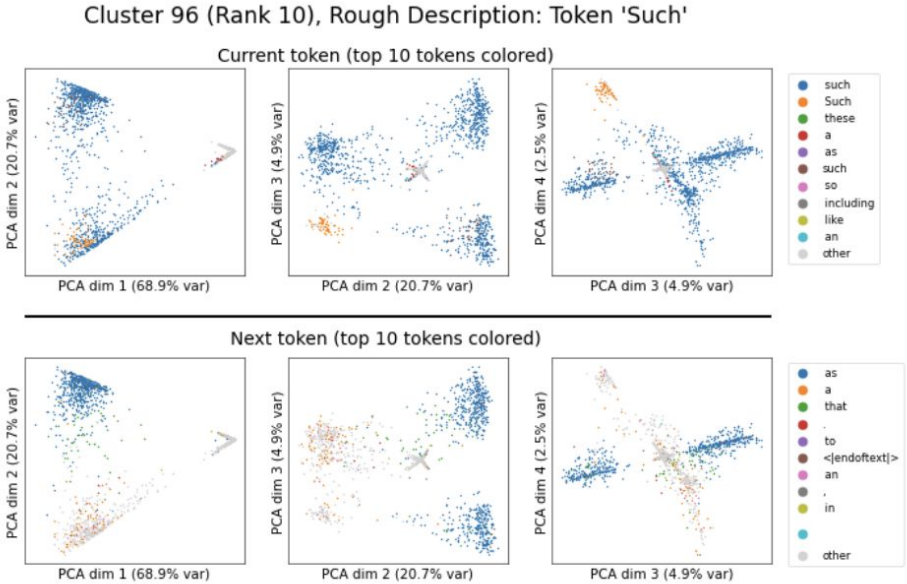


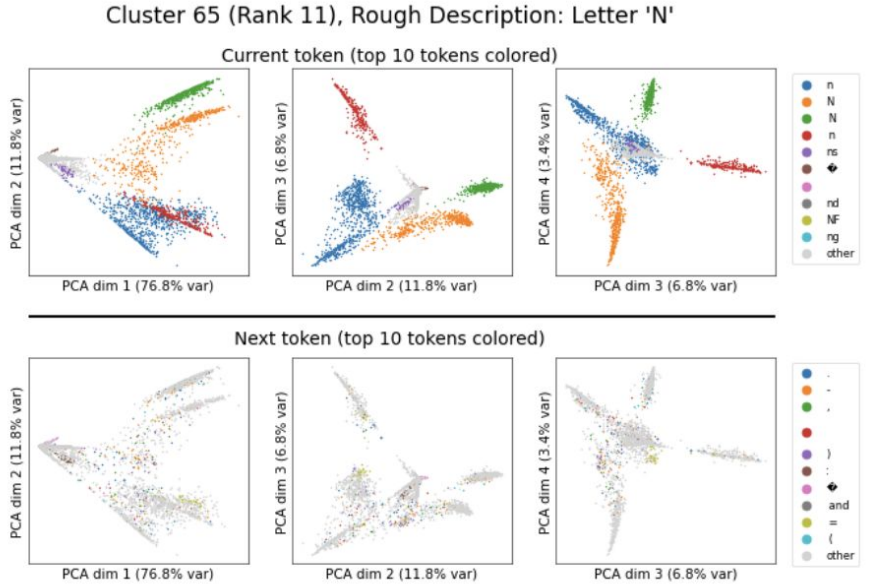
Finding Multi-D Features: Also SAEs

- **Key idea:** many high similarity dictionary vectors will be learned for a given multi-d feature, so *cluster* SAE vectors
- To identify multi-d features, run mixture and separability tests on SAE reconstructions limited to each cluster. This works!

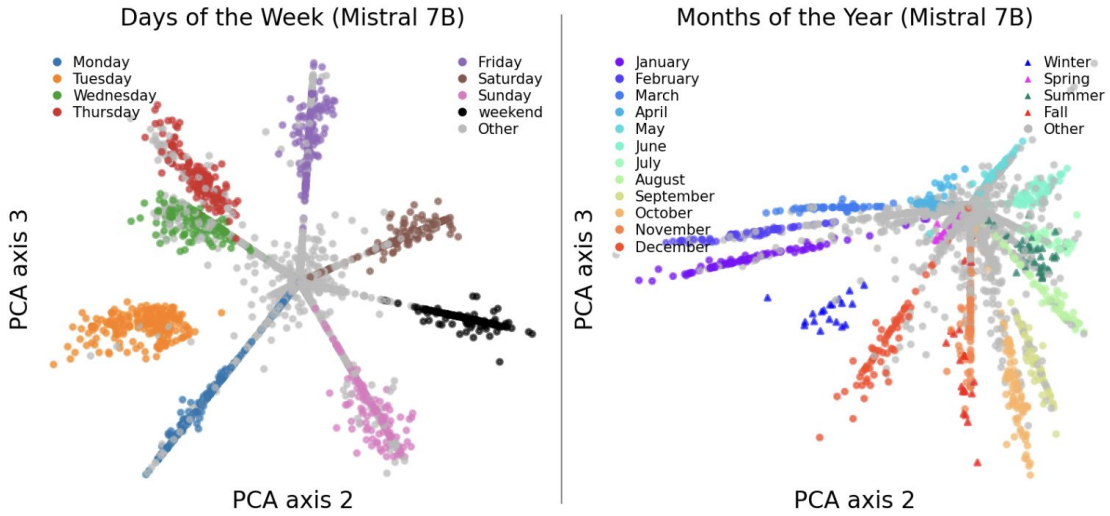


Bonus: What Other Multi-D Features Does Clustering Find?





We Find Some on Mistral 7B Too!



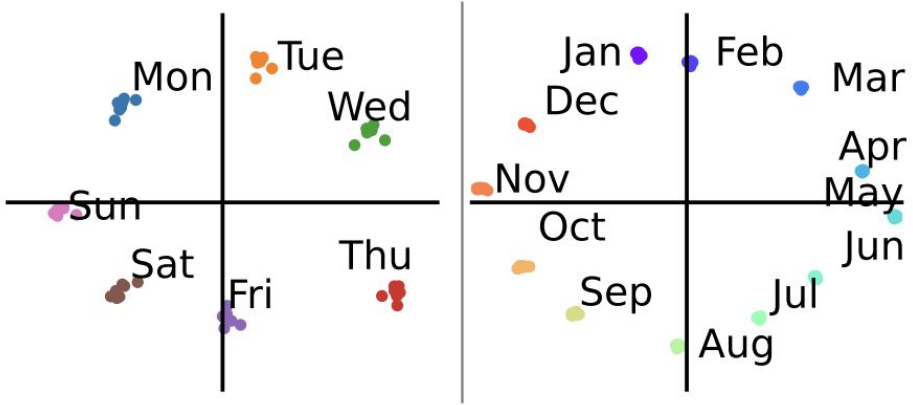
When Do Models Use These Circles: Modular Addition?

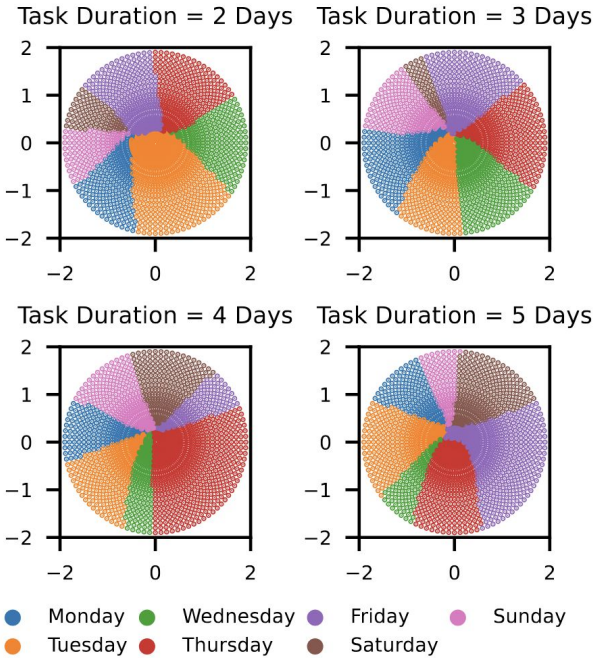
`Weekdays` task: *"Let's do some day of the week math. Two days from Monday is"*
`Months` task: *"Let's do some calendar math. Four months from January is"*

Model	`Weekdays`	`Months`
Llama 3 8B	29 / 49	143 / 144
Mistral 7B	31 / 49	125 / 144
GPT-2	8 / 49	10 / 144

When Do Models Use These Circles: Modular Addition!

- PCA shows models indeed have a circular representation
- We can intervene everywhere in the circle (not just on 7 or 12 learned positions)





Do Models Actually Use These Circles?

- Interventions show they do!
- We train a probe to “fit” the activations with a circle in days of the week, and then intervene on that circle

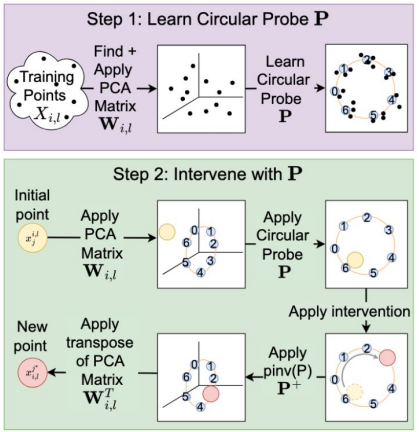


Figure 5: Visual representation of the intervention process.

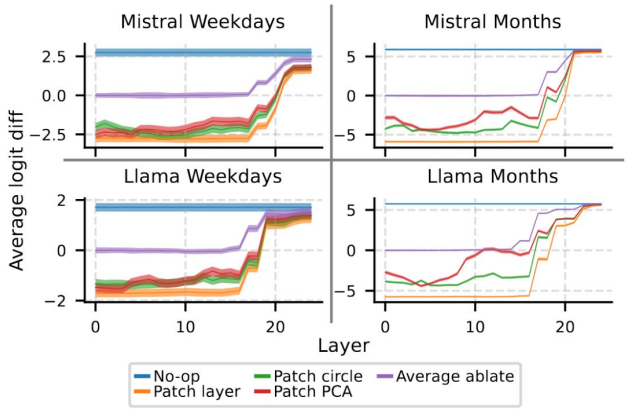
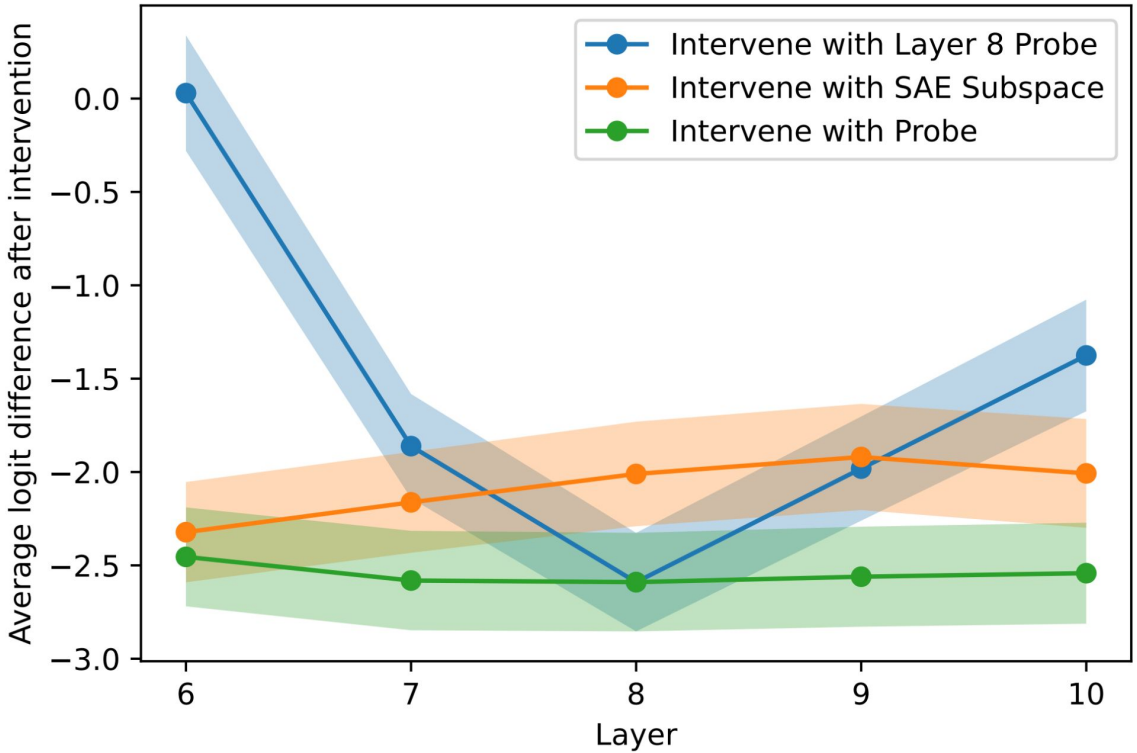
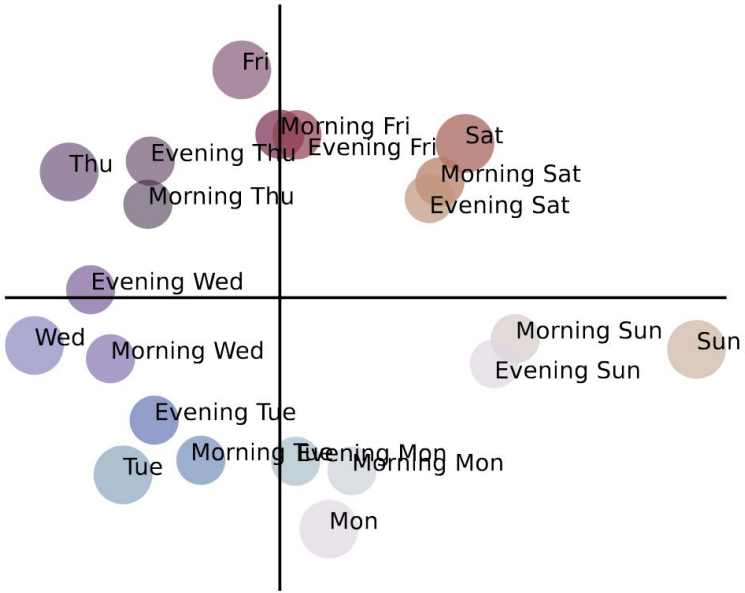


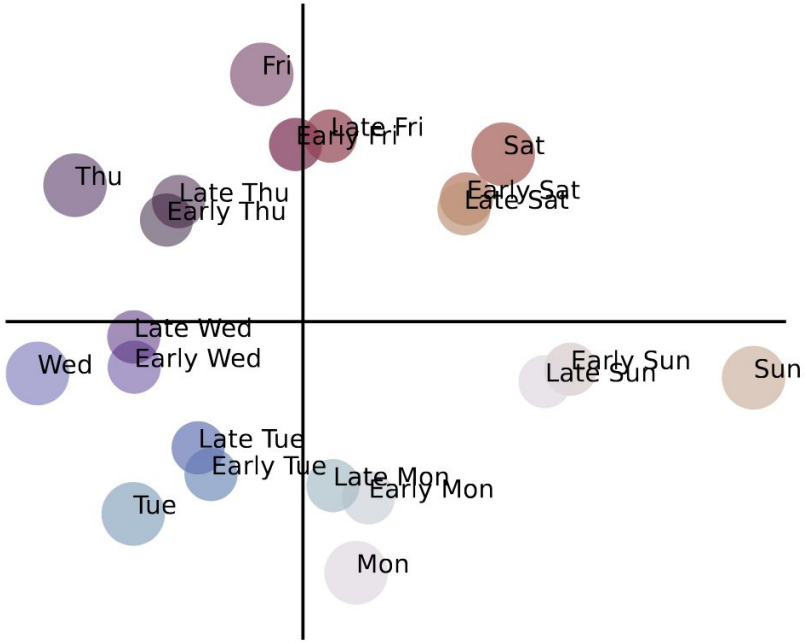
Figure 6: Mean and 96% error bars for intervention methods.

We Can Also Directly Use the Circle From Clustering



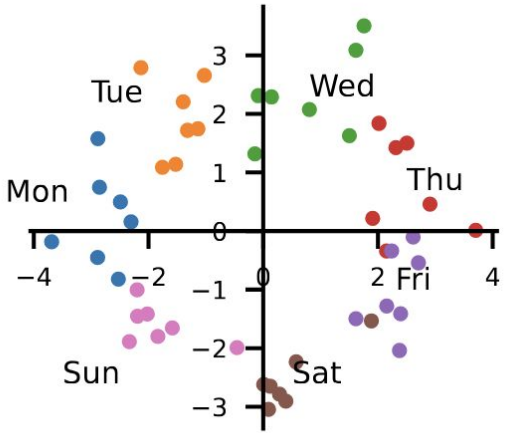
Extra: Circular Representations Are Continuous!





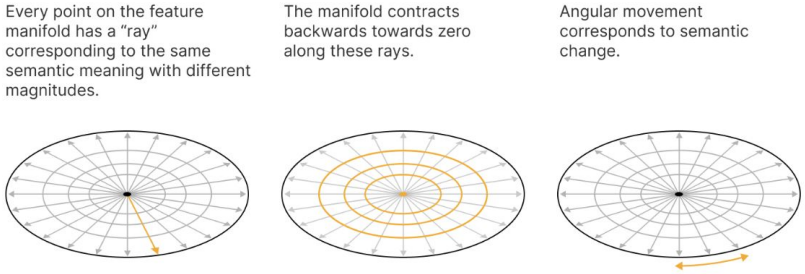
Extra: Discovering Another Circle

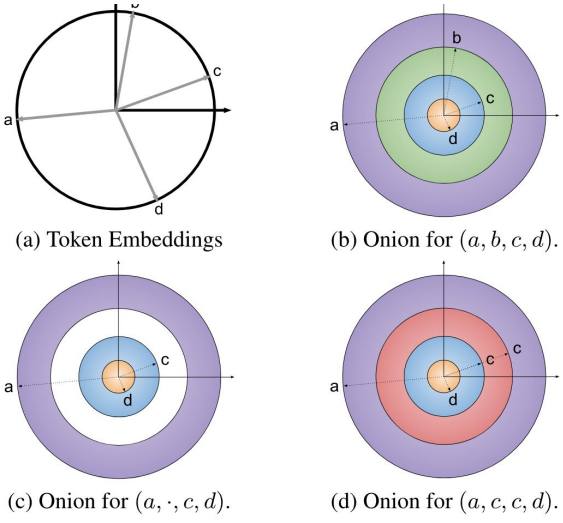
- Method: use regression to remove components from activations that are linearly predictable from start day/month and duration
- In PCA, another circle in the *computed* day/month appears!



Recent Followups: Linear Representation Hypothesis

- Responding to our work, Olah describes a two part LRH:
  - 1. Features are one-dimensional
  - 2. Feature magnitude correlates with intensity
- Csordás et. al. show preliminary evidence against part 2 by finding “onion-like” features in a toy RNN; our evidence is only against part 1





(a) Token Embeddings

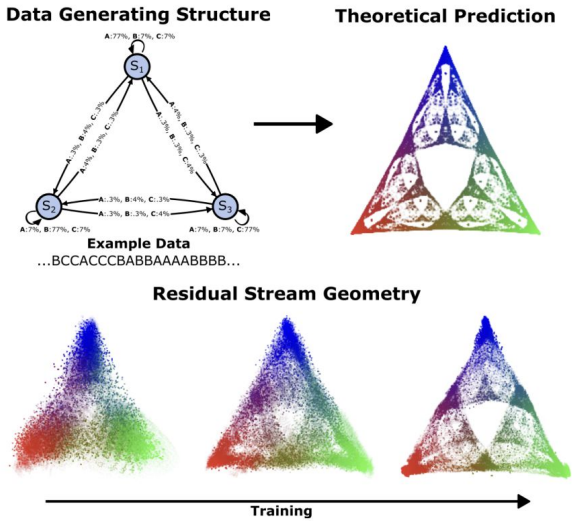
(b) Onion for (a, b, c, d).

(c) Onion for $(a, \cdot, c, d)$.

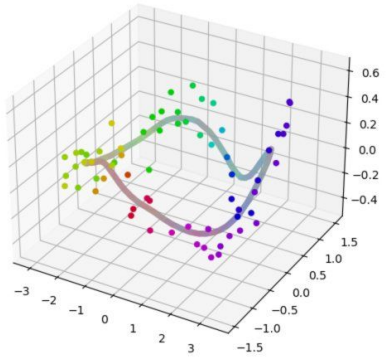
(d) Onion for (a, c, c, d).

Recent Followups: More Multi-Dimensional Features

Belief states in toy models, Shai et. al.



Circular representation for curve detectors, Gorton

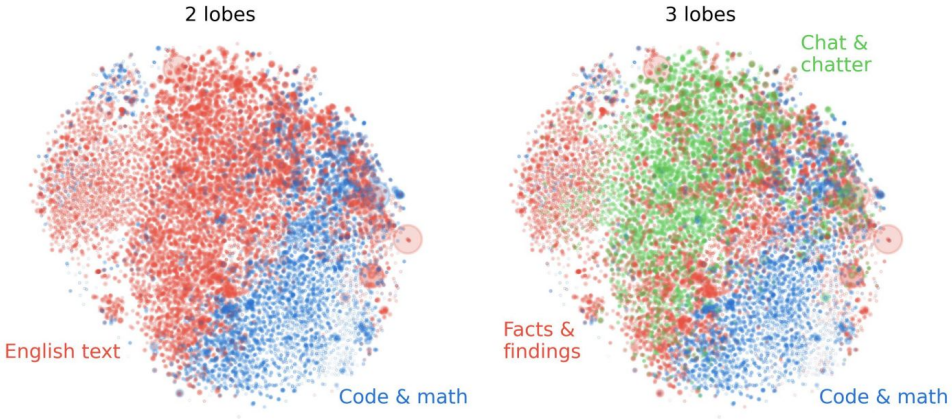


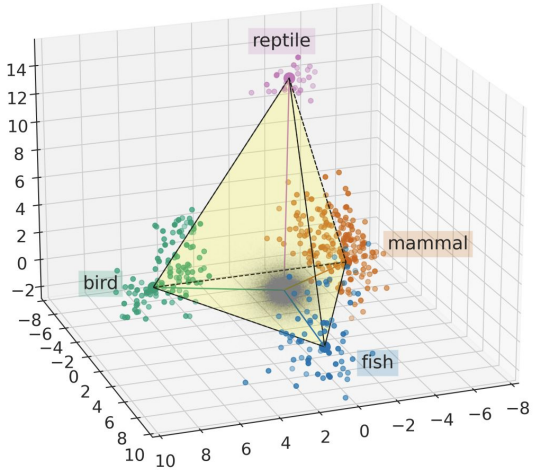
3D PCA of curve features. We observe a circular manifold, but it has “ripples”.

2D PCA of curve features. As curve orientation rotates, we observe a circle.

Recent Followups: Structure in SAEs

- Multiple papers/works on structure in SAEs
  - The Geometry of Concepts: Sparse Autoencoder Feature Structure, Li et. al.
  - SAE feature geometry is outside the superposition hypothesis, Mendel
  - The Geometry of Categorical and Hierarchical Concepts in Large Language Models, Park et. al.





Open Questions on Multi-D Features

- Would be great to find stronger proof that model is really “rotating” circles
- How many features are best thought of as multi-d vs. one-d?
- Can we make SAEs learn multi-d features in the first place?

What Else Might SAEs Be Missing? **Dark Matter**

DECOMPOSING THE DARK MATTER OF SPARSE AUTOENCODERS

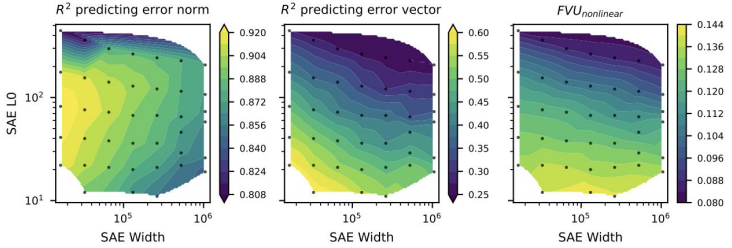
Joshua Engels
MIT
jengels@mit.edu

Logan Smith
Independent
logansmith5@gmail.com

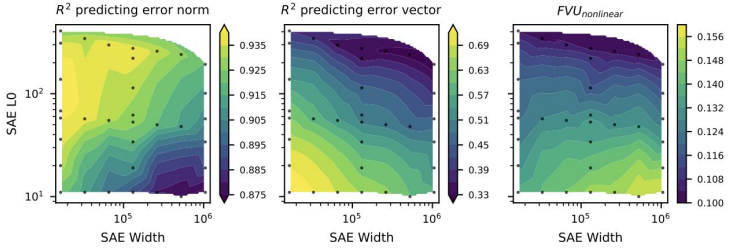
Max Tegmark
MIT & IAIFI
tegmark@mit.edu

SAE Error is Linearly Predictable

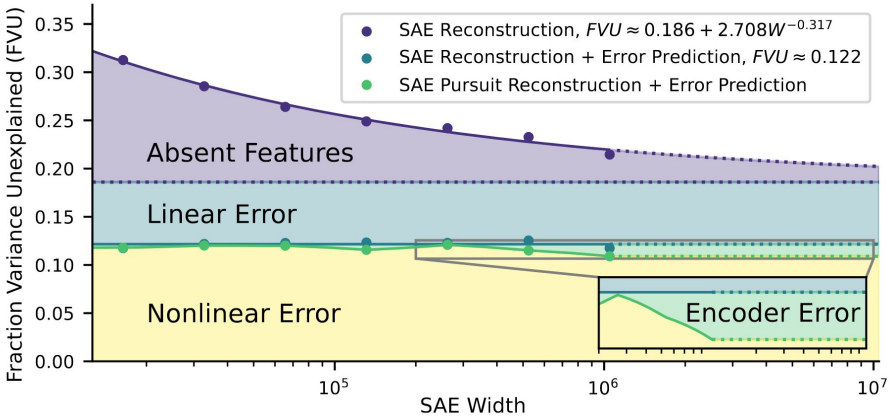
- >90% of SAE error norm and ~50% of error vector is linearly predictable from activations
- Continuing presence of linearly predictable error as we scale = part of activation space we just aren't learning? Are these nonlinear features?



(a) Linear prediction results for layer 12 Gemma 2 2B SAEs from Gemma Scope. $\texttt{FVU}_{\texttt{nonlinear}}$ is roughly constant for a fixed L_0.

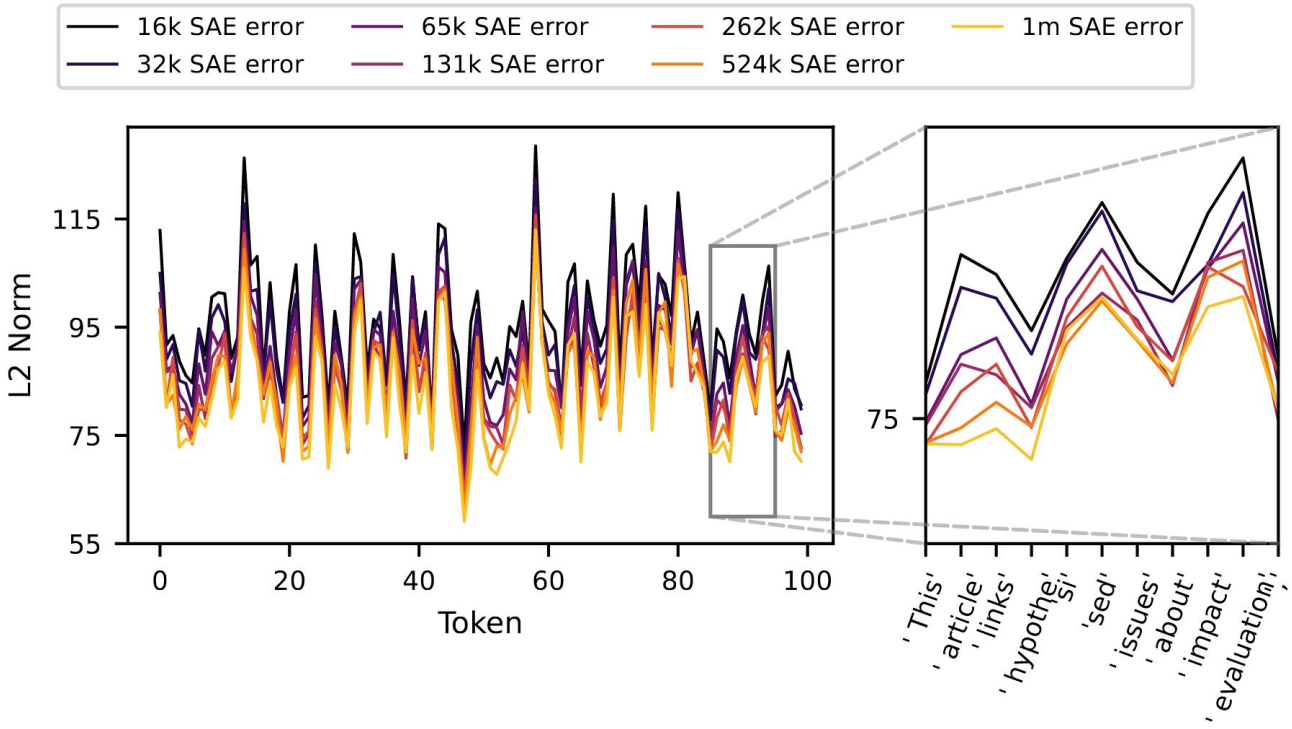


(b) Linear prediction results for layer 20 Gemma 2 9B SAEs from Gemma Scope. $\texttt{FVU}_{\texttt{nonlinear}}$ is roughly constant for a fixed L_0.



SAE Errors Are Consistent Across Scale

- Implies that there are specific contexts that SAEs of any size do badly on?



Nonlinear and Linear Error Components Are Different

- Does nonlinear component contain fewer true model features and more “dense” features?

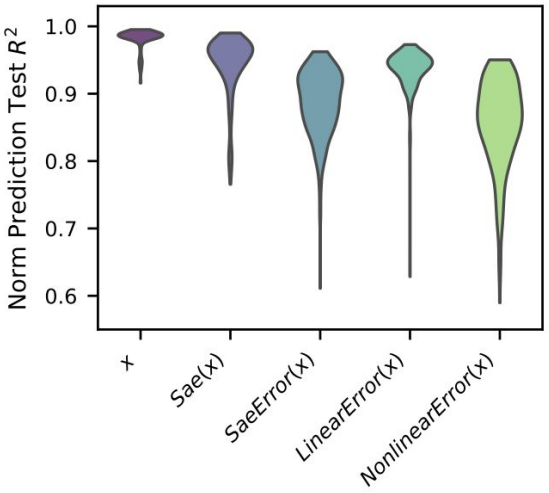


Figure 6: Violin plot of norm prediction tests

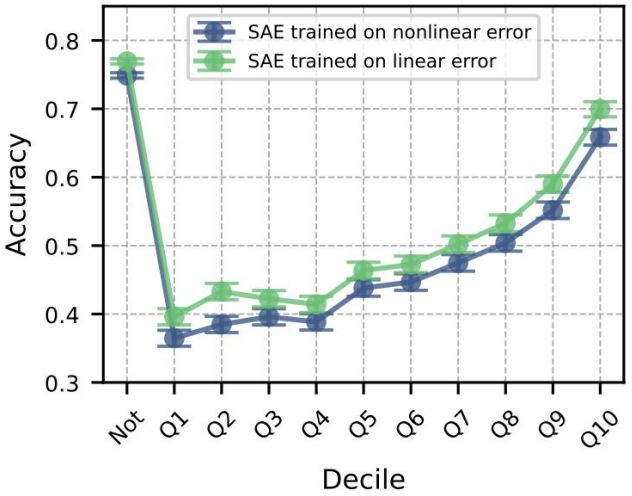
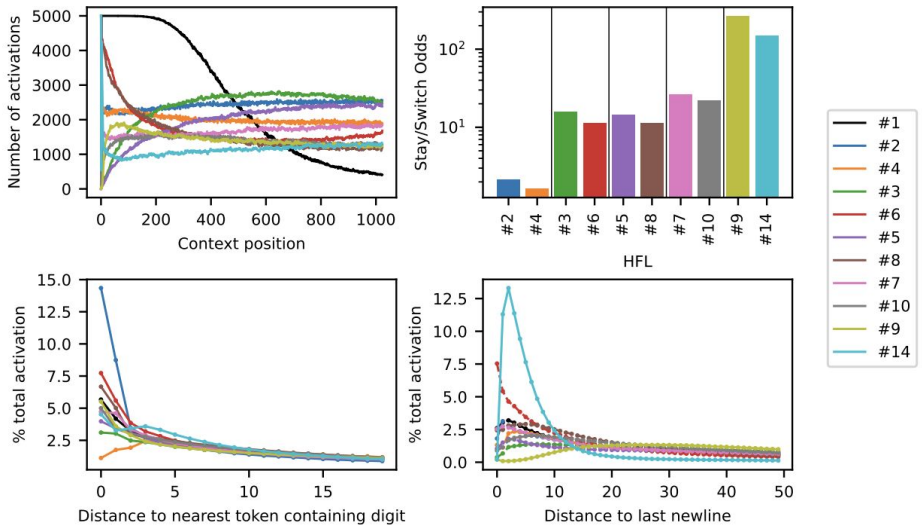


Figure 7: Auto-interpretability results on SAEs

What Else Might SAEs Be Missing? **Dense Features**

000
001
002
003
00
00
00
00

HIGH FREQUENCY LATENTS ARE FEATURES, NOT BUGS



What Else Might SAEs Be Missing? **Usefulness?**

Are Sparse Autoencoders Useful? A Case Study in Sparse Probing

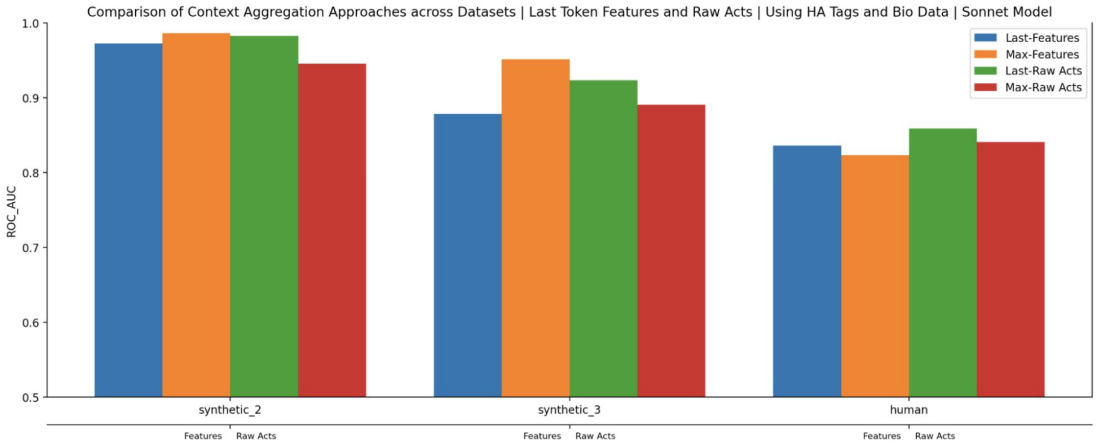
---

Subhash Kantamneni [*1] **Joshua Engels** [*1] **Senthooran Rajamanoharan** **Max Tegmark** [1] **Neel Nanda**




SAEs Should Be Good At Probing

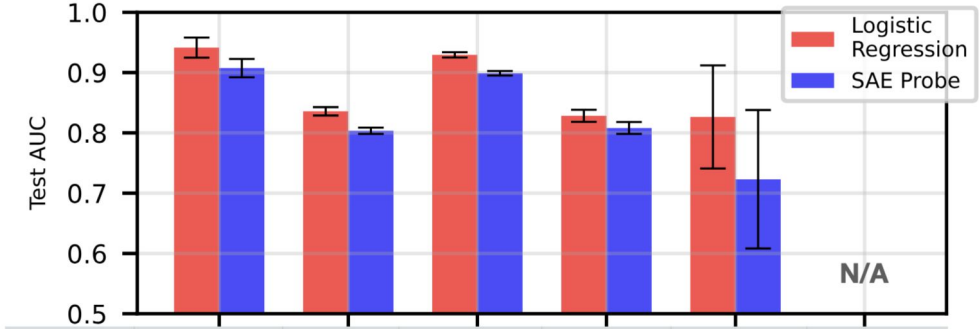
- Probing: logistic regression from model activations to a target
  - Applications: detect deception, honesty, self awareness, etc.
- Intuitively, SAEs might help: inductive bias for more interpretable basis
- Promising prior work from Anthropic



Bricken et. al.

But Are They?

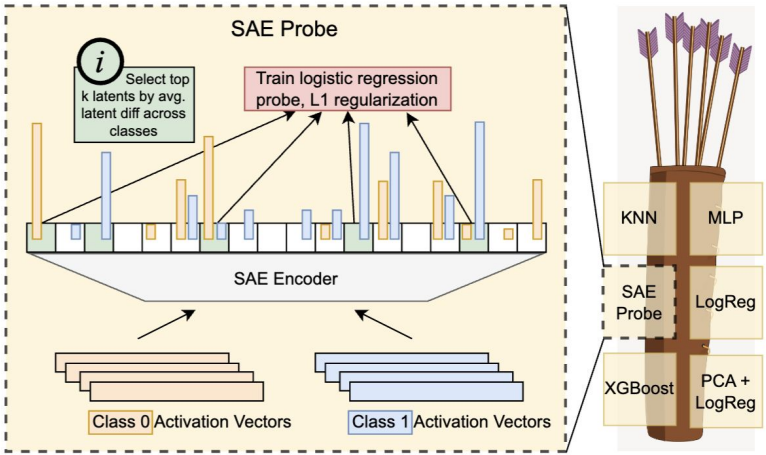
- We investigate on 113 probing datasets
- 5 settings:
  - Standard
  - Data scarcity
  - Class imbalance
  - Noisy labels
  - Out of distribution
- SAE probes perform worse across the board



	Standard Conditions	Data Scarcity	Class Imbalance	Label Noise	OOD	Interp.
Do SAE Probes Help?	✗	✗	✗	✗	✗	~

Can SAE Based Probes At Least Help Sometimes?

- Not really: we try adding SAE-based probes as an option to a simulated practitioner, and taking the best method by val AUC
  - No statically significant improvement



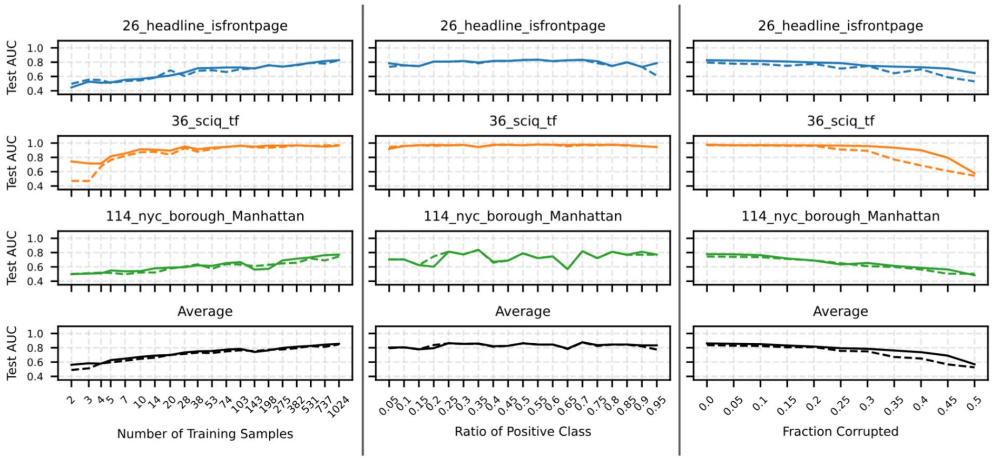


Figure 5. For three of the datasets in Table 1, we visualize the performance when SAE probes are in the quiver (dashed) versus when they are not (solid) for the regimes of data scarcity (left), class imbalance (middle), and label noise (right). In all three regimes, we see that on average (bottom row), SAEs do not help.

What About Those Prior Multi-token Results?

- The advantage partly disappears when using a better baseline of an attention probe

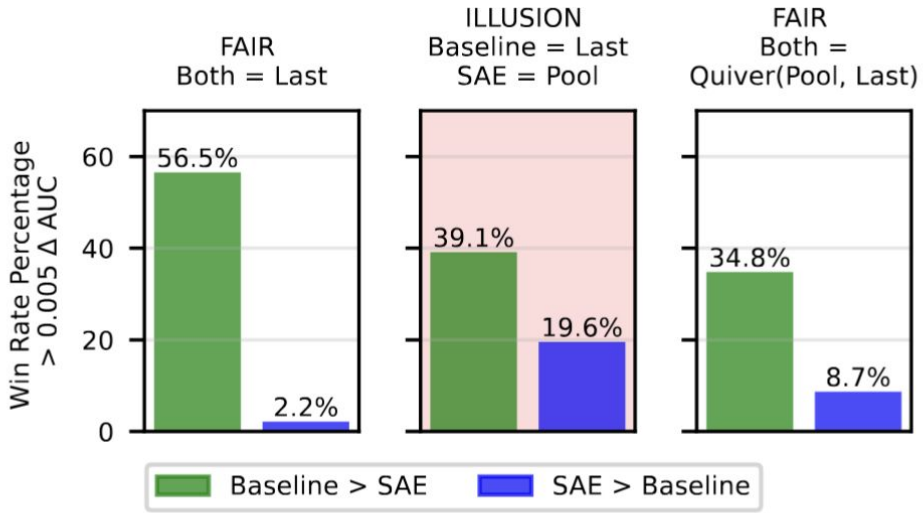


Figure 11. Illustration of a potentially misleading result we find: when comparing activation probes to pooled SAE probes, SAE probe win rate increases considerably, but adding in a "pooled" attention-inspired probe brings down the SAE win-rate.

What About Probe Interpretability?

- SAEs help, but so do baseline methods
- On one dataset, an SAE probe finds a spurious correlation (ending in period)
  - But a logistic regression probe finds this too
- On one dataset, an SAE probe finds mislabeled data
  - But a logistic regression probe finds this too

Table 7. **Left** We show the top 5 examples where latent 369585 is active while the CoLA prompt is labeled as grammatical. Most of these sentences are clearly ungrammatical, which illustrates that the CoLA data is corrupted. **Right** We then look at sentences a baseline classifier trained on CoLA marks as ungrammatical when the label is grammatical. We reach the same conclusion - the CoLA data is mislabeled.

Prompt (Labeled Grammatical)	Latent 369585	Prompt (Labeled Grammatical)	P(y=0)
I don't remember what all I said?	23.09	Rub the cloth on the baby torn.	0.933
Aphrodite said he would free the animals and free the animals he will	15.87	In front of them happen.	0.926
Gilgamesh wanted to seduce Ishtar, and seduce Ishtar he did.	14.92	Who did you give pictures of to friends of?	0.911
An example of these substances be tobacco.	14.85	An example of these substances be tobacco.	0.893
He will can go	14.43	Susan hopes herself to sleep.	0.890

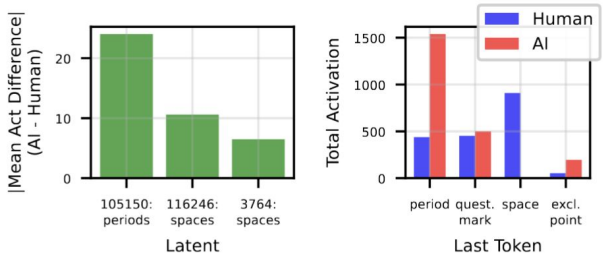


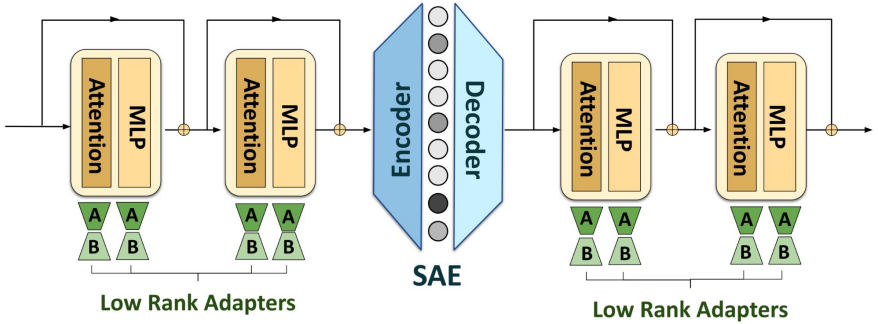
Figure 10. **Left:** Top 3 latents (w/ descriptions) by mean activation difference between AI generated and human SAE latent activations. **Right:** Histograms of the identity of the 4 most common last tokens in AI generated and human text.

How Might We Improve SAEs to Fix These Problems?

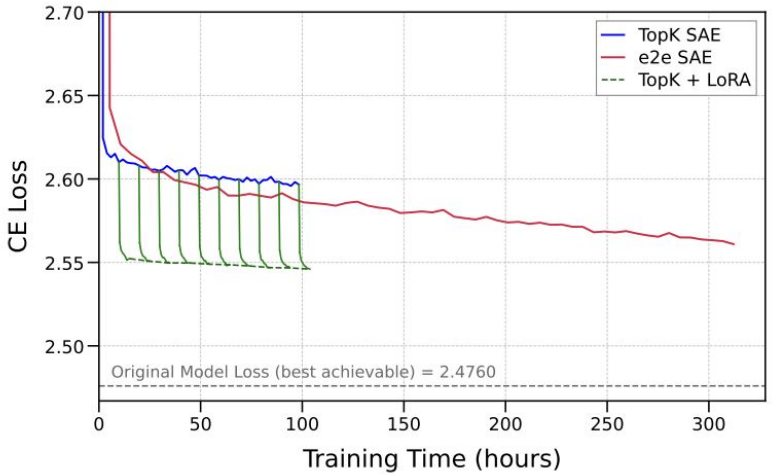
One idea: Improve the model + SAE system

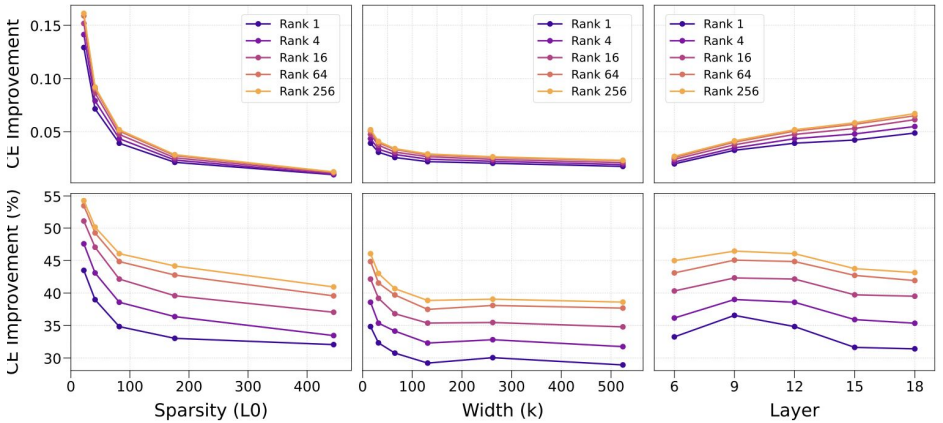
Low-Rank Adapting Models for Sparse Autoencoders

Matthew Chen [*1] Joshua Engels [*1] Max Tegmark [1]



Reduces up to 60% of the Loss Gap to No Sae





Also Helps Downstream Metrics

Downstream Metric	TopK + LoRA	TopK
SCR (max)	**0.526**	0.448
SCR (average)	**0.314**	0.289
TPP (max)	**0.412**	0.372
TPP (average)	**0.145**	0.111
Sparse Probing (top 1)	**0.760**	0.732
Sparse Probing (test)	**0.956**	0.955
Autointerp	0.830	**0.832**
Absorption	0.210	**0.205**

Some Speculation: What Does the Future Hold for Saes?

- Are they useful for real model auditing and control tasks?
- What auxiliary losses can we add to get “better” latents?
- How can we make automated interpretability *way* better?
- Do we need transcoders / cross layer approaches? Parameter based decomposition?

Questions