Measuring Progress in Dictionary Learning for Language Model Interpretability with Board Game Models


Adam Karvonen *


Benjamin Wright *


Can Rager


Rico Angell


Jannik Brinkmann

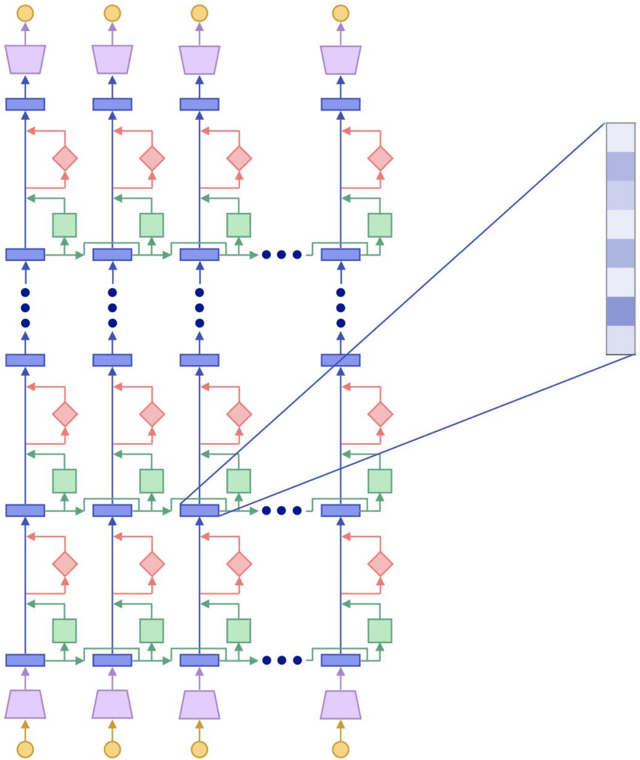

Logan Smith


Claudio Mayrink Verdun

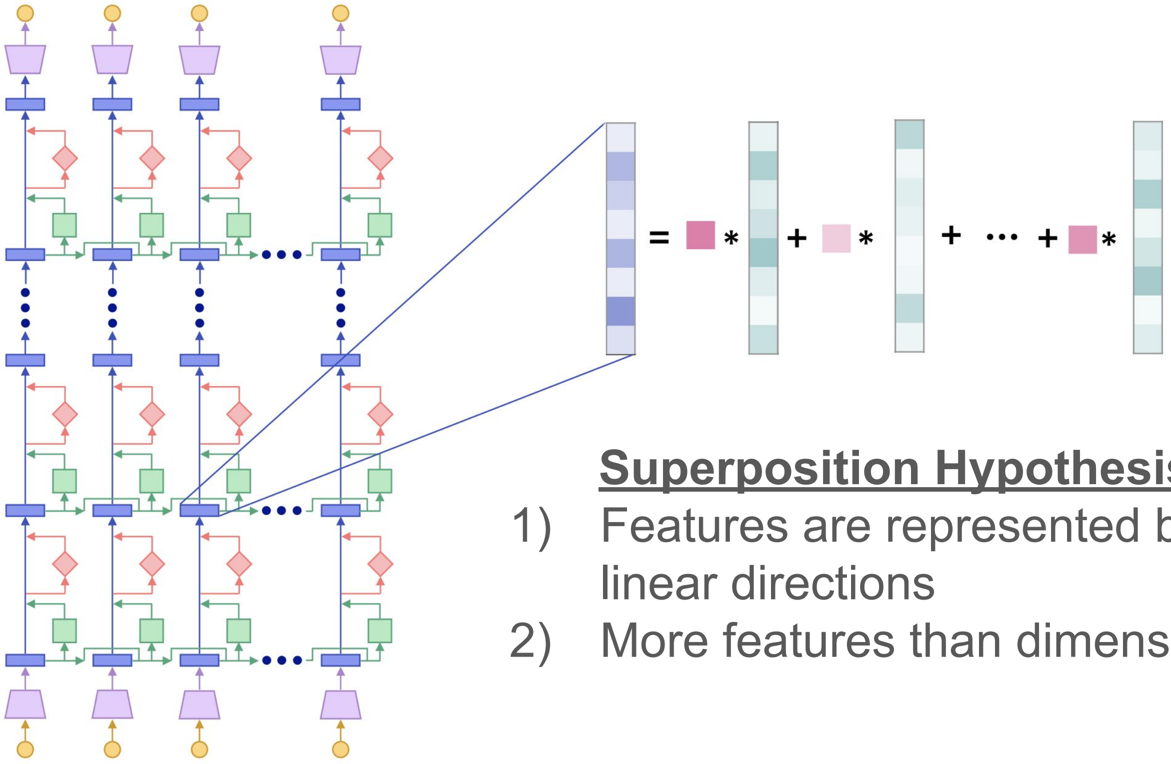

David Bau


Samuel Marks

Fundamental Units of Representation



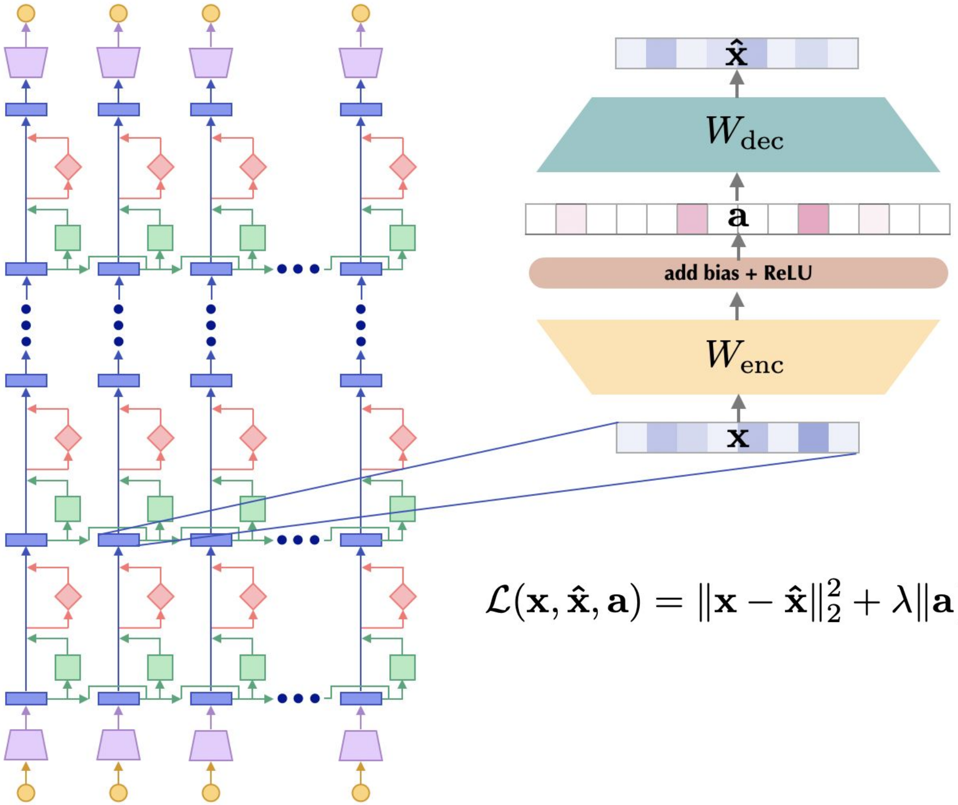
Fundamental Units of Representation



Superposition Hypothesis

1) Features are represented by linear directions
2) More features than dimensions

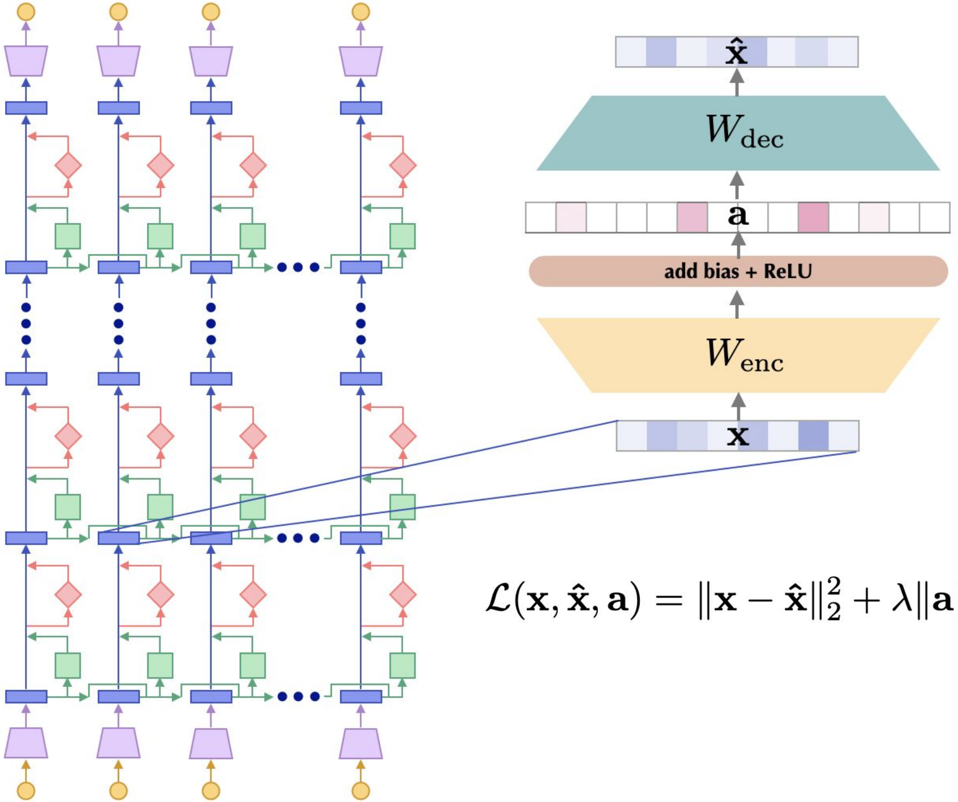
Standard SAE Training



$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{a}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \lambda \|\mathbf{a}\|_1$$

Train sparse autoencoder (SAE) to reconstruct activations using only a few feature vectors

Standard SAE Training

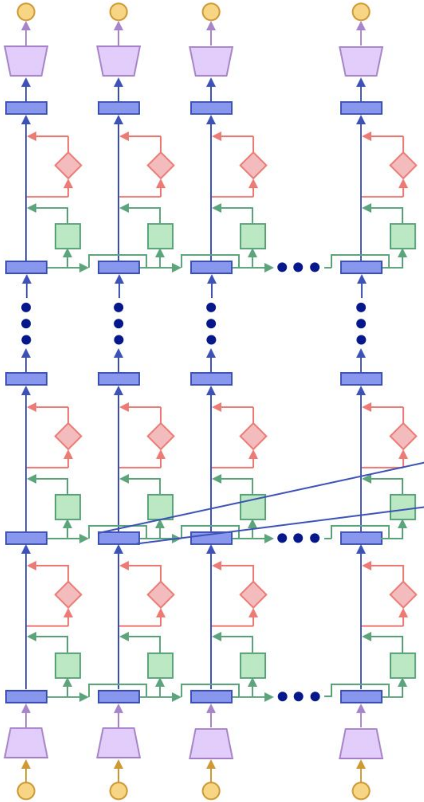


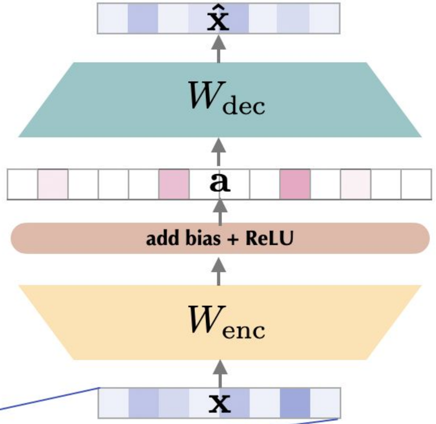
$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{a}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \lambda \|\mathbf{a}\|_1$$

Train sparse autoencoder (SAE) to reconstruct activations using only a few feature vectors

- \+ Simple and efficient

Standard SAE Training



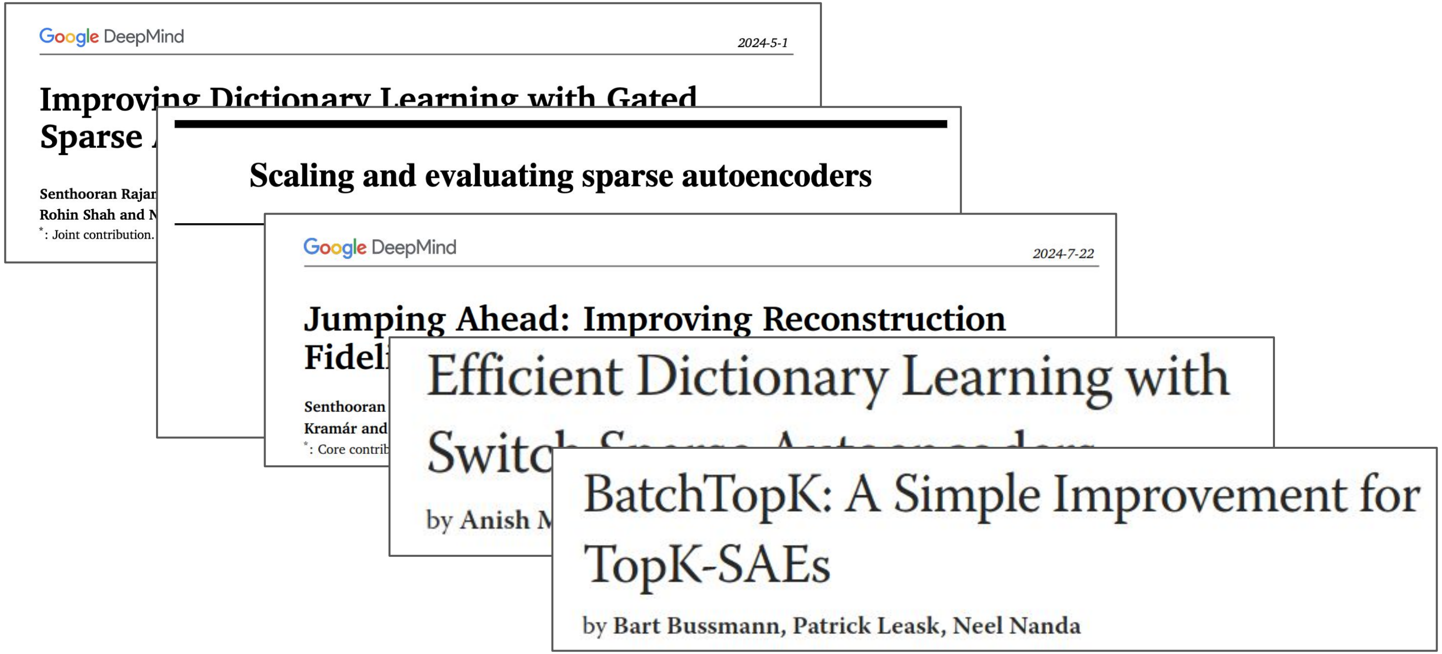


$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{a}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \lambda \|\mathbf{a}\|_1$$

Train sparse autoencoder (SAE) to reconstruct activations using only a few feature vectors

- + Simple and efficient
- **+ Entirely unsupervised!**

Many new SAEs proposed (or rediscovered)



Google DeepMind 2024-5-1

Improving Dictionary Learning with Gated Sparse

Senthooran Rajar
Rohin Shah and N
*: Joint contribution.

Scaling and evaluating sparse autoencoders

Google DeepMind 2024-7-22

Jumping Ahead: Improving Reconstruction Fideli

Senthooran
Kramár and
*: Core contrib

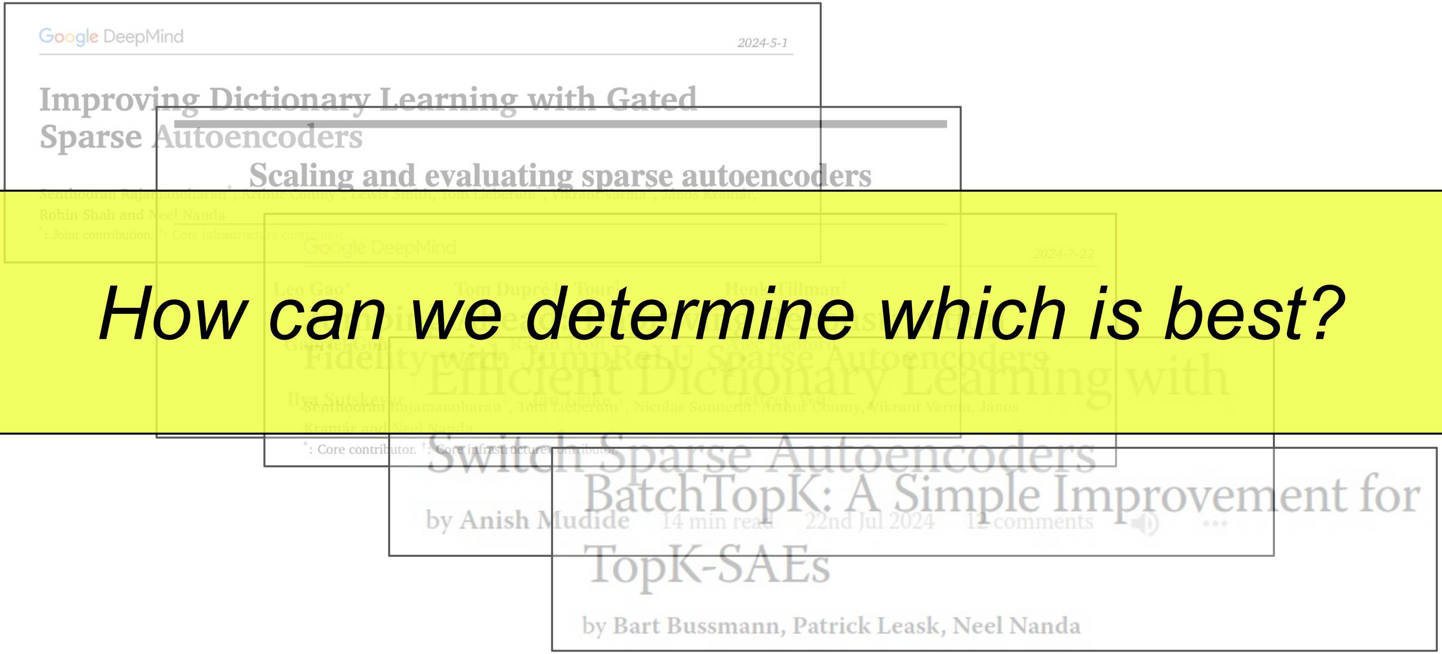
Efficient Dictionary Learning with Switc

by Anish M

BatchTopK: A Simple Improvement for TopK-SAEs

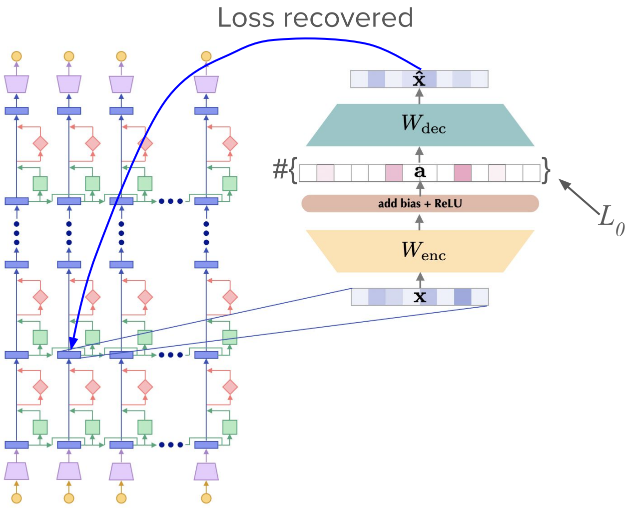
by Bart Bussmann, Patrick Leask, Neel Nanda

Many new SAEs proposed (or rediscovered)



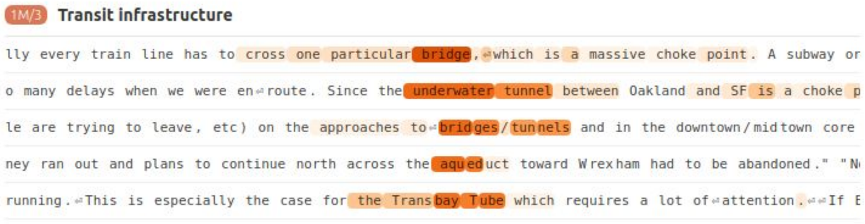
How can we determine which is best?

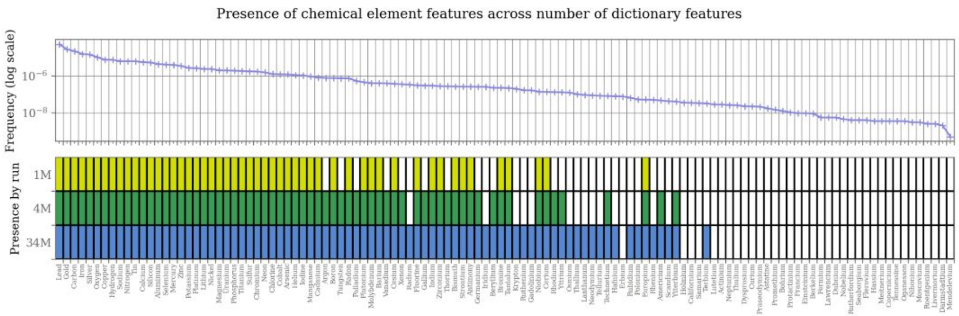
Current Evaluation Strategies



Unsupervised Proxy Metrics

Manual Inspection





Weakly-Supervised Metrics

Board Games

… have an **explicit** board state.

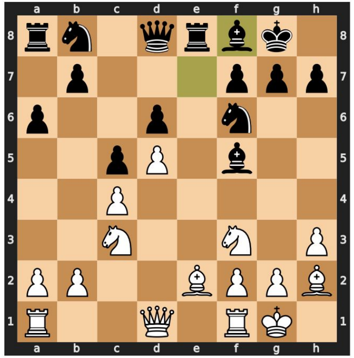
… easily enumerable feature space.

… board states are **objective.**

Board Game Language Models

Consider chess GPT, trained to predict the next character in transcripts of real chess games

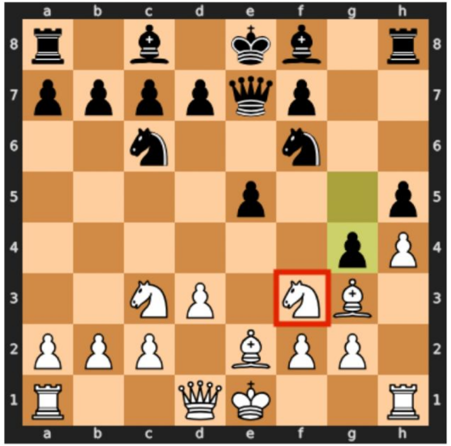
```
1.c4 Nf6 2.Nc3 c5 3.d4 e6 4.d5 d6
5.e4 exd5 6.exd5 Be7 7.Bf4 O-O
8.Be2 a6 9.Nf3 Bd7 10.O-O Re8
11.h3 Bf5 12.Bh2 Bf8
```



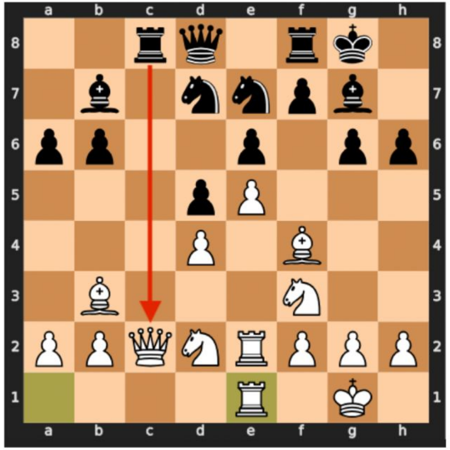
We can extract ***explicit*** and ***deterministic*** board states from the model

We also consider OthelloGPT as a much simpler game

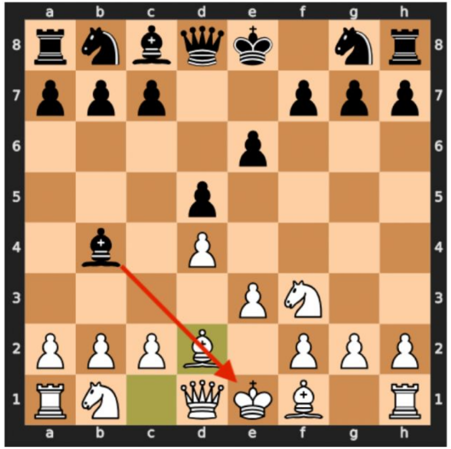
Board State Properties (BSPs)



knight on `f3`



rook threat present

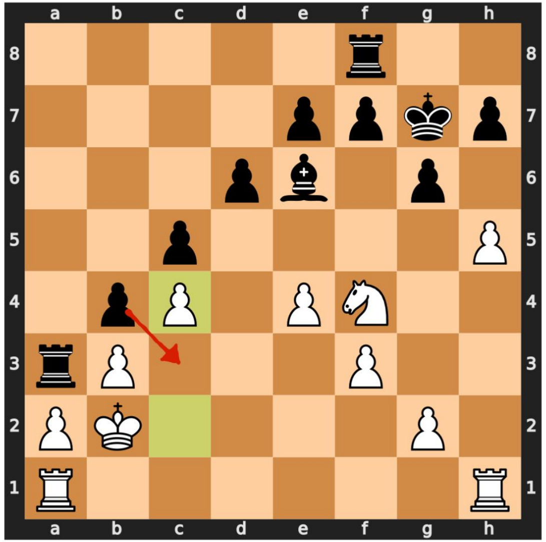


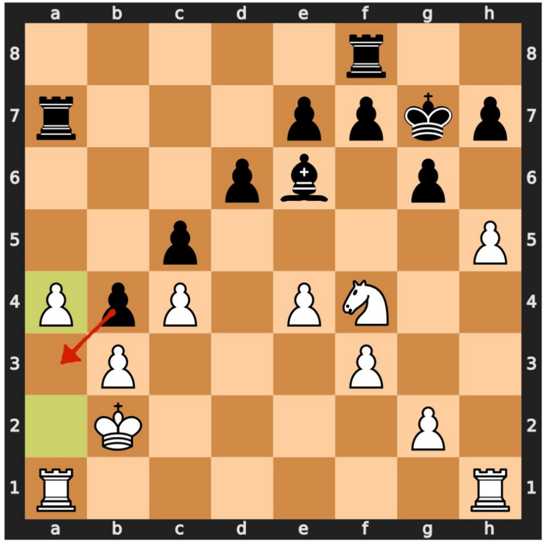
pin present

We evaluate ~1000 BSPs varying from low-level board states to high-level strategy

Then, we automatically find features connected to BSPs

SAE feature representing *en passant*





1.e4 c5 2.Nc3 Nc6 3.Nf3 g6 4.d4 cxd4 5.Nxd4 Bg7 6.Be3 Nf6 7.Qd2 Ng4 8.Nxc6 bxc6 9.Bd4 Bxd4 10.Qxd4 0-0 11.Be2 d6 12.Bxg4 Bxg4 13.f3 Be6 14.h4 Qb6 15.0-0-0 Rab8 16.Qxb6 axb6 17.h5 Kg7 18.b3 b5 19.Kb2 b4 20.Ne2 c5 21.Nf4 Ra8 22.Ra1 Ra3 23.c4 Ra7 24.a4

Contribution #1: Board Game Metrics

Using our BSP's, we construct two ***supervised*** SAE metrics:

1) **Coverage:** How well do features align with individual BSPs?

2) **Board Reconstruction:** How well can we reconstruct the board given SAE features?

Coverage

How well do features align with individual BSPs?

BSP	Max F1 score of any SAE feature
White Pawn on B6	0.99
White Pawn on B7	0.83
…	…
Black Queen on H7	0.23
Average	**0.63**

Board Reconstruction

How well can we reconstruct the board given SAE features?

1. **Identify high precision SAE features**
(i.e. when the feature is active, the BSP is present)

Feature	is high-precision for
# 0	White Pawn on B6
# 1	None
#2	White Pawn on B6
…	…
# N	Black Queen on D5 and White Pawn on D4

Board Reconstruction

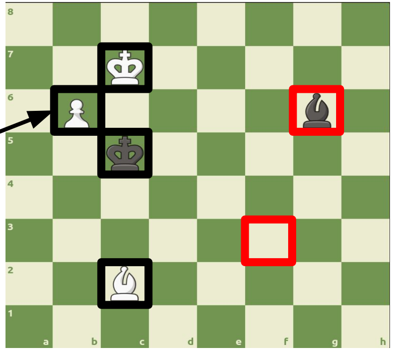
How well can we reconstruct the board given SAE features?

1. **Identify high precision SAE features**
(i.e. when the feature is active, the BSP is present)

2. **Reconstruct board state based on feature activations**
(on an unseen test game)

Feature	is high-precision for
# 0	White Pawn on B6
# 1	None
#2	White Pawn on B6
…	…
# N	Black Queen on D5 and White Pawn on D4

Reconstruction



Board Reconstruction

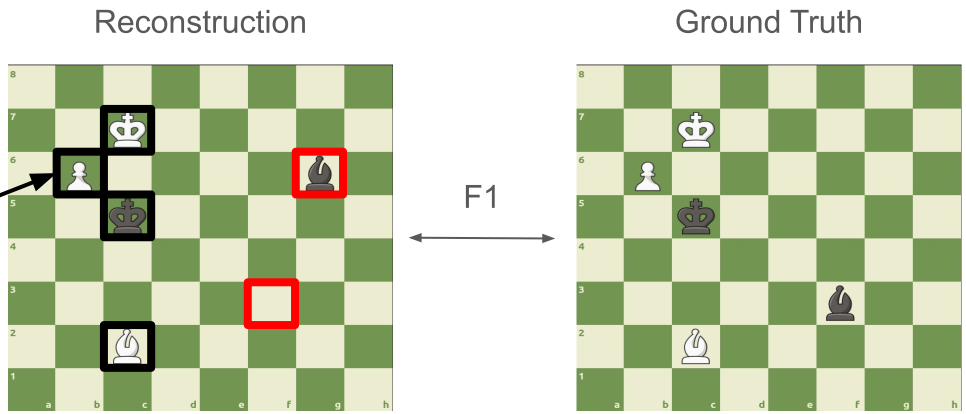
How well can we reconstruct the board given SAE features?

1. **Identify high precision SAE features** (i.e. when the feature is active, the BSP is present)

2. **Reconstruct board state based on feature activations** (on an unseen test game)

3. **Compare to ground truth** by calculating F1-score

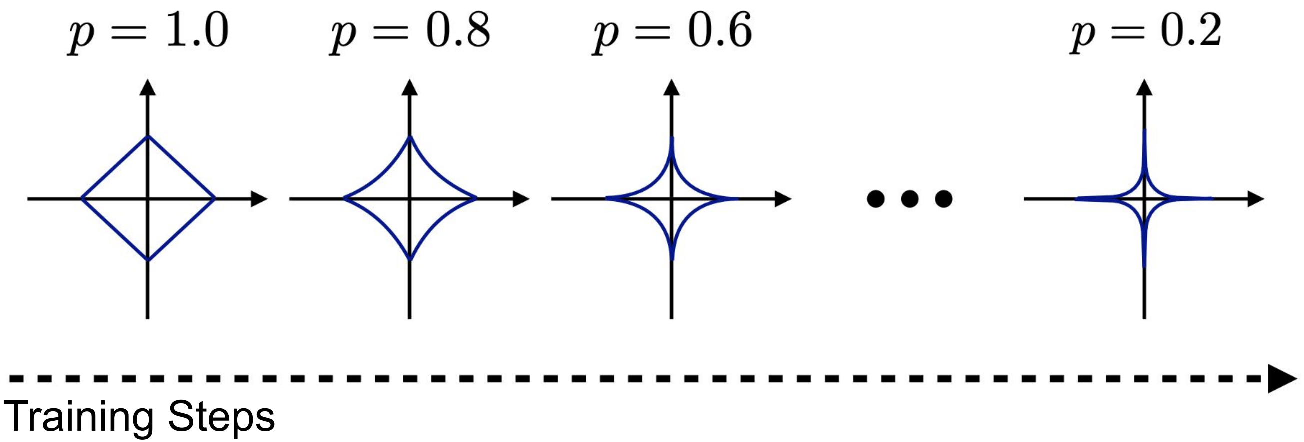
Feature	is high-precision for
# 0	White Pawn on B6
# 1	None
#2	White Pawn on B6
…	…
# N	Black Queen on D5 and White Pawn on D4



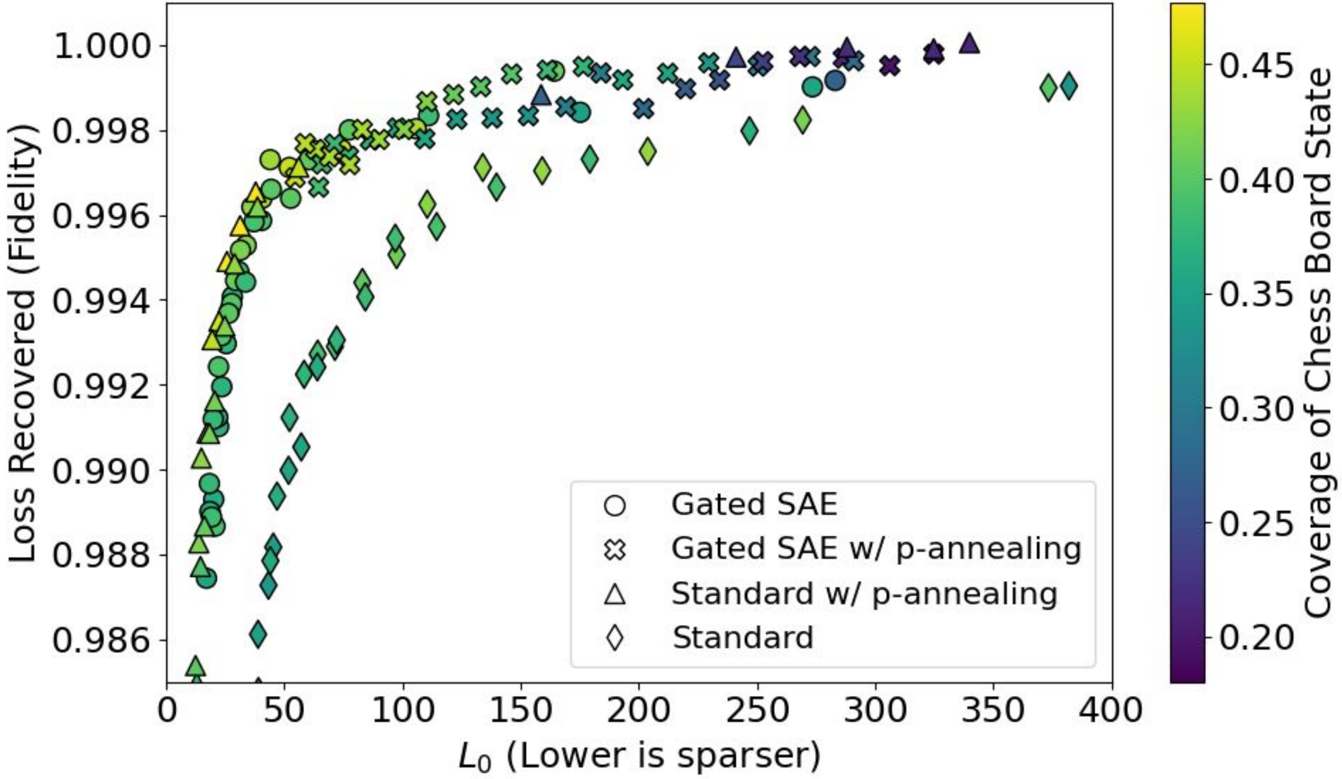
Contribution #2: p-Annealing SAE training technique

<u>Idea:</u> Replace L_1-norm minimization with $L_p^{\,p}$-norm, anneal p during training

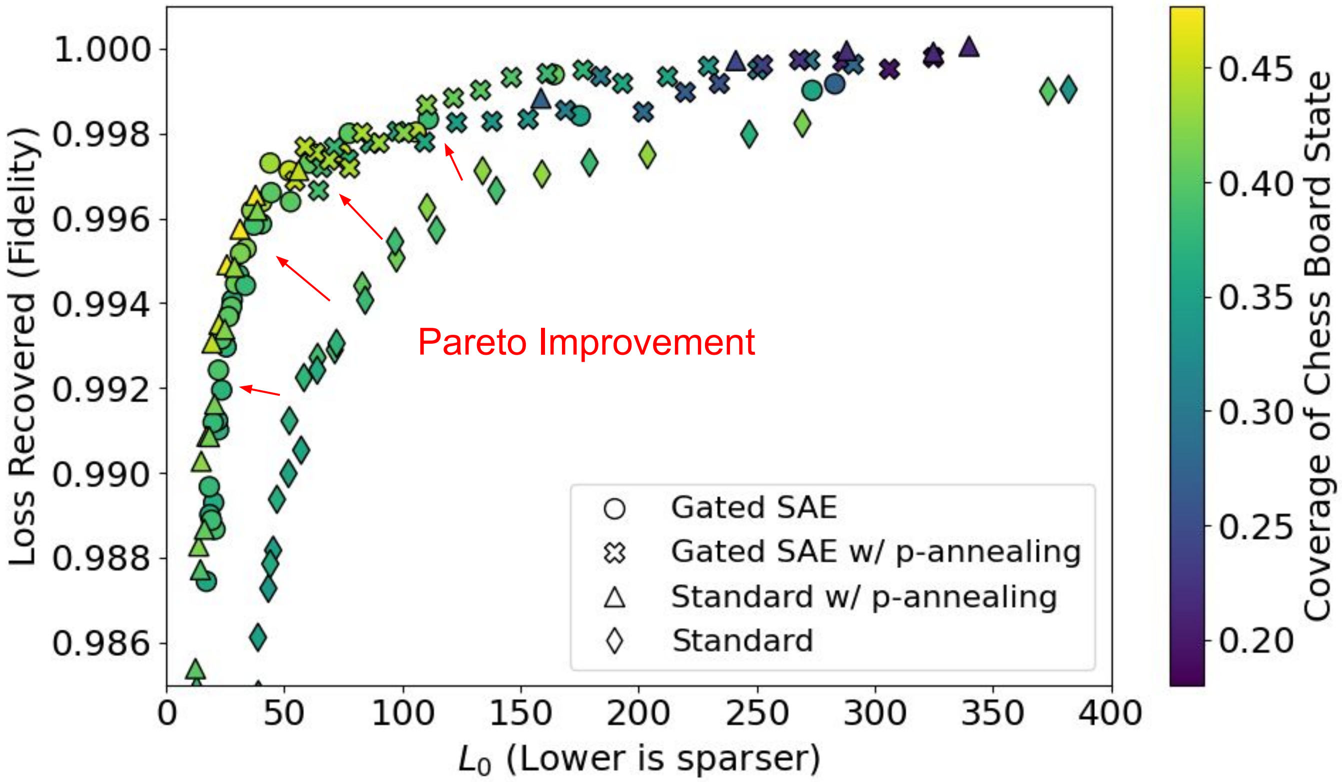
$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{a}, p) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \lambda \|\mathbf{a}\|_p^p$$



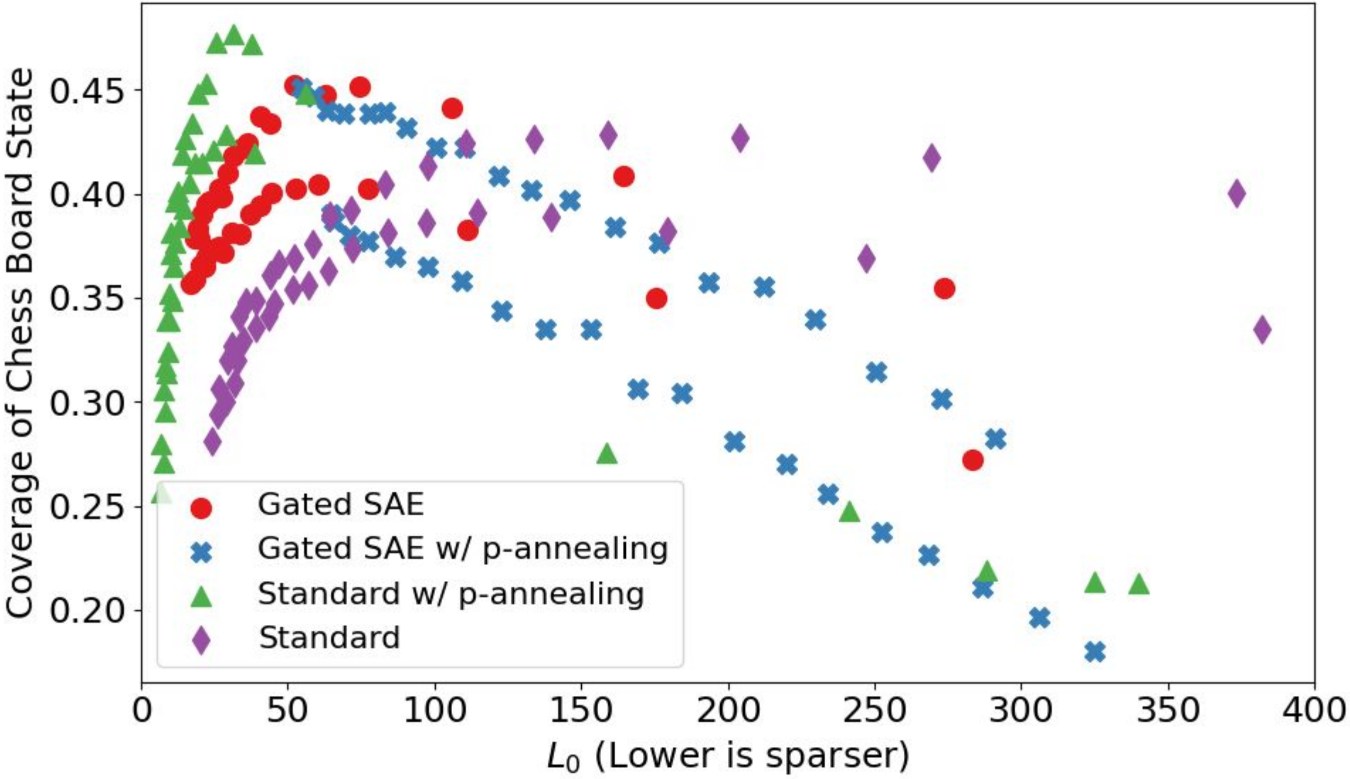
BSP Metrics Correlate with SAE Quality



BSP Metrics Correlate with SAE Quality



BSP Metrics Can Differentiate Between SAE Architectures



Conclusions and Future Work

How can we compare different SAEs?

What fraction of the GPT's world model do the SAEs capture?

Future work:

1. Create better evaluations for natural language
2. Further understand board game models