Thomas Heap 1 Tim Lawson 1 Lucy Farnik 1 Laurence Aitchison 1

Abstract

Sparse autoencoders (SAEs) are an increasingly popular technique for interpreting the internal representations of transformers. In this paper, we apply SAEs to 'interpret' random transformers, i.e., transformers where the parameters are sampled IID from a Gaussian rather than trained on text data. We find that random and trained transformers produce similarly interpretable SAE latents, and we confirm this finding quantitatively using an open-source auto-interpretability pipeline. Further, we find that SAE quality metrics are broadly similar for random and trained transformers. We find that these results hold across model sizes and layers. We discuss a number of number interesting questions that this work raises for the use of SAEs and auto-interpretability in the context of mechanistic interpretability.

1. Introduction

Sparse autoencoders (SAEs) have become a popular tool in mechanistic interpretability research, with the aim of disentangling the internal representations of neural networks by learning sparse, interpretable features from network activations (Elhage et al., 2022; Sharkey et al., 2022; Cunningham et al., 2023; Bricken et al., 2023). An autoencoder with a high-dimensional hidden layer is trained to reconstruct activations while enforcing sparsity (Gao et al., 2024; Templeton et al., 2024; Lieberum et al., 2024), with the aim of discovering the underlying concepts or 'features' learned by the network (Park et al., 2023; Wattenberg & Viégas, 2024).

An interpretability technique that aims to understand the computations a model has learned to perform during training should distinguish between trained and untrained models or models whose parameters have been randomized. We show that SAEs do not always achieve this, suggesting they provide limited insights into the computational structure learned by the underlying model. In particular, we found that the

latents learned by SAEs trained on transformers whose parameters are randomized have similar auto-interpretability scores (Bills et al., 2023; Paulo et al., 2024) to the latents learned by SAEs trained on the activations of the corresponding unmodified transformer. Additionally, we found that randomized transformers are much closer to trained models than to our control in terms of standard SAE evaluations.

This result raises important questions about what we can glean from applying SAEs to interpret language models. While SAEs extract interpretable features (e.g., Cunningham et al., 2023; Bricken et al., 2023; Rajamanoharan et al., 2024a), our findings suggest that many features are likely to reflect statistical properties of the training data (Dooms & Wilhelm, 2024) and underlying model architecture rather than learned computations. This challenges the widespread interest in understanding the mechanisms of trained models with SAEs (Sharkey et al., 2025), highlighting the need to develop techniques that better capture computations.

These results have important implications for mechanistic interpretability research. In particular, we suggest that more rigorous methods to distinguish between artifacts and genuinely learned computations are needed, and that interpretability techniques should be carefully validated against appropriate null models.

Finally, we speculate about why these patterns might emerge. At a high level, there are two hypotheses:

- 1. The input data is already superposed, and randomly initialized NNs largely preserve this superposition.
- 2. Randomly initialized NNs amplify or even introduce superposed structure to the input data (e.g., given dense input generated IID from a Gaussian).

We present toy models to demonstrate the plausibility of these hypotheses in Section 4 but defer conclusions as to the mechanism responsible to future work.

2. Related Work

Sparse dictionary learning Under a different name, 'superposition' in visual data is one of the foundational observations of computational neuroscience. Olshausen & Field (1996; 1997) showed that the receptive fields of simple cells in the mammalian visual cortex can be explained as a result

¹University of Bristol, Bristol, UK. Correspondence to: Thomas Heap < thomas.heap@bristol.ac.uk>.

of sparse coding, i.e., representing a relatively large number of signals (sensory information) by simultaneously activating a relatively small number of elements (neurons). Coding theory offers a perspective on efforts to extract the 'underlying signals' responsible for neural network activations (Marshall & Kirchner, 2024).

Sparse dictionary learning (SDL) approximates a set of input vectors by linear combinations of a relatively small number of learned basis vectors. The learned basis is usually overcomplete: it has a greater dimension than the inputs. SDL algorithms include Independent Component Analysis (ICA), which finds a linear representation of the data such that the components are maximally statistically independent (Bell & Sejnowski, 1995; Hyvärinen & Oja, 2000). Sparse autoencoders (SAEs) are a simple neural network approach (Lee et al., 2006; Ng, 2011; Makhzani & Frey, 2014). Typically, an autoencoder with a single hidden layer that is many times larger than the input activation vectors is trained with an objective that imposes or incentivizes sparsity in its hidden layer activations to try to find this structure.

Mechanistic interpretability Recently, it has become common to understand 'features' or concepts in language models as low-dimensional subspaces of internal model activations (Park et al., 2023; Wattenberg & Viégas, 2024; Engels et al., 2024). If such sparse or 'superposed' structure exists, we expect to be able to 'intervene on' or 'steer' the activations, i.e., to modify or replace them to express different concepts and so influence model behavior (Meng et al., 2022; Zhang & Nanda, 2023; Heimersheim & Nanda, 2024; Makelov, 2024; O'Brien et al., 2024).

SAEs are a popular approach to discovering these features, where we typically train a single autoencoder to reconstruct the activations of a single neural network layer, e.g., the transformer residual stream (Sharkey et al., 2022; Cunningham et al., 2023; Bricken et al., 2023). A wide variety of SAE architectures have been suggested, which commonly vary the activation function applied after the linear encoder (Makhzani & Frey, 2014; Gao et al., 2024; Rajamanoharan et al., 2024b; Lieberum et al., 2024). SAEs have also been trained with different objectives (Braun et al., 2024) and applied to multiple layers simultaneously (Yun et al., 2021; Lawson et al., 2024; Lindsey et al., 2024).

Besides reconstruction errors and preservation of the underlying model's performance, SAEs have been evaluated according to whether they capture specific concepts (Gurnee et al., 2023; Gao et al., 2024) or factual knowledge (Huang et al., 2024; Chaudhary & Geiger, 2024), and whether these can be used to 'unlearn' concepts (Karvonen et al., 2024b).

Automatic neuron description Sparse autoencoders often learn tens of thousands of latents, which are infeasi-

ble to describe by hand. Yun et al. (2021) find the tokens that maximally activate a dictionary element from a text dataset and manually inspect activation patterns. Instead, researchers typically collect latent activation patterns over a text dataset and prompt a large language model to explain them (e.g. Bills et al., 2023; Foote et al., 2023). These methods have been widely adopted (e.g. Cunningham et al., 2023; Bricken et al., 2023; Gao et al., 2024; Templeton et al., 2024; Lieberum et al., 2024).

Bills et al. (2023) generate an explanation for the activation patterns of a language-model neuron over examples from a dataset, simulate the patterns based on the explanation, and score the explanation by comparing the observed and simulated activations. This method is commonly known as autointerpretability (as in self-interpreting). Paulo et al. (2024) introduce classification-based measures of the fidelity of automatic descriptions that are inexpensive to compute relative to simulating activation patterns and an open-source pipeline to compute these measures. Choi et al. (2024) use best-of-*k* sampling to generate multiple explanations based on different subsets of the examples that maximally activate a neuron. Importantly, they fine-tune Llama-3.1-8B-Instruct on the top-scoring explanations to obtain inexpensive 'explainer' and 'simulator' models.

Polysemanticity Lecomte et al. (2024) point out that neurons may become polysemantic incidentally. Polysemanticity is the notion that a single neuron (basis dimension) of a network layer may represent multiple interpretable concepts (Elhage et al., 2022; Scherlis et al., 2023); unsurprisingly, individual neurons in a randomly initialized network may be polysemantic (Lecomte et al., 2024). By contrast, our work studies *superposition* (Elhage et al., 2022; Chan, 2024), which pertains to the representations learned across a whole network layer as opposed to any individual neuron. In particular, superposition allows a network layer as a whole to represent a larger number of (sparse) features than the layer has (dense) neurons by sparse coding (only a few concepts are active at a time, i.e., a given token position).

Training only the embeddings Recently, Zhong & Andreas (2024) demonstrated some surprising capabilities of transformers where only the embeddings are trained and no other parameters. These results demonstrate that the behavior of a randomly initialized transformer can be shaped to a surprising extent by training only a few parameters. However, this is very different from our setting, not only in that we consider SAEs, but also in that our 'Step-O' and 'Re-randomized incl. embeddings' variants randomize all the parameters, including the embeddings. While the 'Re-randomized excl. embeddings' variant does use pre-trained embeddings, we do not train those embeddings with fixed, randomized weights but simply freeze the pre-trained em-

beddings and randomize the other weights (Section 3).

3. Results

We trained per-layer SAEs on the residual stream activation vectors of transformer language models from the Pythia suite with between 70 million and 7 billion parameters (Biderman et al., 2023). We compared SAEs trained on different variants of the underlying transformers:

Trained: The usual, trained model.

Re-randomized incl. embeddings: All the model parameters, including the embeddings, are re-initialized by sampling Gaussian noise with mean and variance equal to the values for each of the original, trained weight matrices.

Re-randomized excl. embeddings: As above, except the embedding and unembedding weight matrices are not reinitialized, i.e., are the same as the original, trained model.

Step-0: For Pythia models, the step0 revisions are available, which are the original model weights at initialization, i.e., before any learning (Biderman et al., 2023).

Control: The original, trained model, except where the input token embeddings are replaced at inference time by sampling IID standard Gaussian noise for each token, such that a given token does not have a consistent embedding vector. We expect auto-interpretability to perform at the level of chance for this variant.

We train on 100 million tokens from the RedPajama dataset (Weber et al., 2024; TogetherComputer, 2023) with an activation buffer size of 10 million tokens. For models with fewer than 410 million parameters, we trained an SAE at every layer; for Pythia-1B, at every second layer; and for Pythia-6.9B, at every fourth layer.

Unless otherwise stated, we trained k-sparse autoencoders, a.k.a. TopK SAEs, with an expansion factor of R=64 and k=32. We confirm that our results are robust with respect to these hyperparameters by training SAEs on Pythia-160M with expansion factors equal to powers of 2 between 16 and 128 and k values of 16 and 32 (Figure 11).

Our training implementation is based on Belrose (2024); our evaluation implementations are based on EleutherAI and Karvonen et al. (2024a).

3.1. Auto-interpretability

Feature explanations, which identify some concept, can be used as input to a classifier that predicts the presence of that concept within a piece of text. We can evaluate such a classifier using traditional metrics like the area under the receiver operating characteristic curve (AUROC). There is a choice of classification tasks for evaluating feature explanations,

e.g., fuzzing and detection scoring (Paulo et al., 2024).

We use the 'fuzzing' scoring method, where both positive and negative examples of tokens (i.e., with non-zero and zero activation values, respectively) for a given latent are delimited with special characters, and a language model is prompted to identify which examples have been correctly delimited for the latent given its explanation. By contrast, simulation scoring is based on the correlation between simulated and observed activations (Bills et al., 2023).

For each trained SAE (i.e., underlying model, variant, and layer), we randomly sampled 100 features to obtain auto-interpretability scores. The implementation is based on Paulo et al. (2024). We use the Meta-Llama-3.1-70B-Instruct-AWQ-INT4 model to generate explanations and make predictions, larger than the 8B models used by Choi et al. (2024) and opensource unlike Bills et al. (2023).

We found that the auto-interpretability scores were far more similar between the trained and randomized models than with the control (Figures 1, 2, and 3). The similarity between the ROC scores for trained and randomized transformers demonstrates that 'fuzzing' auto-interpretability, applied to SAE latent explanations, fails to meaningfully distinguish between these underlying models.

3.2. Evaluation

We went on to consider standard SAE quality metrics, along with the auto-interpretability AUROC for Pythia models with between 70M and 6.9B parameters. As above, we broadly found that the randomly initialized and rerandomized models (Figure 3 blue, green, orange lines) were more similar to the trained model (Figure 3 gray lines) than to our control (Figure 3 black lines).

Notably, the cosine similarity and explained variance are often far lower for the random control than the other models, and its reconstruction errors tend to increase across layers while the remaining variants decrease. For the random control, this can perhaps be explained by the fact that a Gaussian is the highest entropy distribution with fixed mean and variance (Jaynes, 2003); we speculate that Gaussian vectors are the 'least structured', in some sense, and thus hardest for SAEs to reconstruct. As Gaussian-distributed activations are propagated through successive layers, we would expect the activations to become less Gaussian and perhaps more 'sparse', i.e., easier to reconstruct (Section 4).

Interestingly, the randomized variants (blue and orange lines) are more similar to the trained model than the variant at initialization (green line). This is especially evident if we look at the L^1 norm values in larger models. We speculate that this pattern arises because parameter norms may differ greatly between a trained model and its state

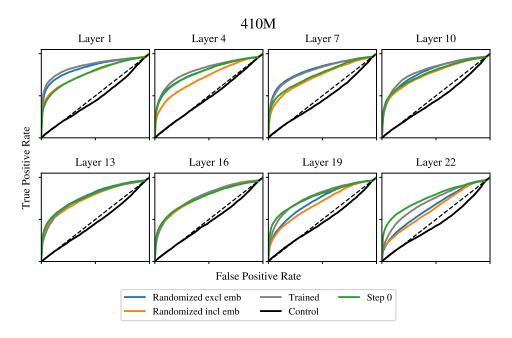


Figure 1. ROC curves for 'fuzzing' auto-interpretability for Pythia-410M over 100 SAE latents. These results demonstrate the similarity in performance between the SAE variants, as well as the overall degradation in performance as the layer index increases. The auto-interpretability scores here fail to distinguish between trained and randomized models.

at initialization. In contrast, our randomization procedure was specifically designed to preserve parameter norms with respect to the trained model. The scale of parameters at different layers may be important, e.g., to control the growth of activations as they progress through the residual stream (Liu et al., 2020). In the AUROC plots, we find that for all but the control variant, AUROC increases with model size. We speculate that features become more specific as the size of the SAEs increases: in smaller SAEs, each latent has to explain more of the input, making classification tasks easier.

Figure 3 (fifth row) shows the cross-entropy (CE) loss score, a.k.a. the loss recovered, broken down by layer. This is the increase in the CE loss when the original model activations are replaced by their SAE reconstructions, divided by the increase in the CE loss when the activations are replaced by zeros ('ablated'). The results show that the 'trained' variant SAEs perform similarly to others from the literature (e.g., Kissane et al., 2024; Rajamanoharan et al., 2024a; Mudide et al., 2024). Importantly, the CE loss score only makes sense for the trained variant: for any of the randomized variants, the CE loss is very poor regardless of whether the original or reconstructed activations are used.

3.3. Latent explanation complexity

While randomized and trained transformers achieve similar auto-interpretability scores and evaluation metrics, we suspected that SAEs applied to these very different settings might capture qualitatively different features. In particular, we suspected that SAEs trained on the randomized variants could learn simple features depending primarily on characteristics of the input text (Dooms & Wilhelm, 2024) but not more complex, abstract features, like those captured with trained transformers (e.g. Templeton et al., 2024). Given the difficulty of qualitatively evaluating features at scale, we attempt a quantitative evaluation in terms of the distribution of latent activations over token IDs. We provide a random sample of features and the corresponding maximally activating dataset examples for each SAE variant of Pythia-6.9B in Appendix D.

Anecdotally, we have observed that a significant proportion of SAE latents have non-zero activations only on a single token or a small number of distinct tokens within a text dataset (e.g., Lin & Bloom, 2024; Dooms & Wilhelm, 2024). Hence, a simple measure of the complexity of an SAE latent given a set of maximally activating examples is the degree to which the latent activates on a single token ID or multiple distinct token IDs. Specifically, we quantify the number of token IDs in terms of the entropy of the observed distribution of latent activations over tokens (vocabulary items): the greater the entropy, the more 'spread out' the latent activations over the vocabulary, and the less token-specific the latent. We take this distribution to be the total latent activation per token across the set of maximally activating examples used to generate explanations for auto-interpretability.

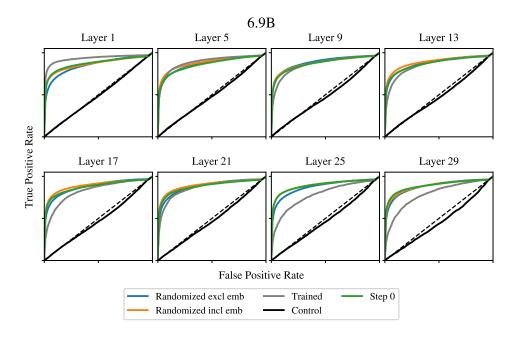


Figure 2. ROC curves for 'fuzzing' auto-interpretability for Pythia-6.9B over 100 SAE latents. Similarly to Figure 1, these results demonstrate the similarity in performance between the SAE variants variants. Notably, the trained variant appears to degrade in performance for the later model layers.

We include the entropy of the observed distributions of latent activations over token IDs in the last row of Figure 3. The negative control variant displays a consistently high entropy, which is to be expected given that the embedding for a given token ID is sampled IID from a Gaussian on each occurrence of the token, i.e., a token does not have a consistent embedding vector (Section 3). For the trained variant, the entropy increases across layers, i.e., the further into the model, the less likely the maximally activating examples for each latent contain activations concentrated on a single token. This is also expected: at later layers, we expect more abstract features that are less similar to token embeddings. Finally, the entropy for randomized models tends to be lower than either the trained or control variants, indicating that latents are activated specifically at a single token ID or a small number of IDs.

In combination with the preceding results, this suggests that standard SAE quality and auto-interpretability metrics are missing an important aspect of SAE features: their "abstractness". While the token distribution entropy is not a direct measure of "abstractness", it suggests that the randomized variants, viewed in the context of their similar auto-interpretability scores to the trained variant, remain able to learn simple, single-token features. However, unlike the trained variant, the features of the randomized variants do not become more complex as the layer index increases.

4. A toy model of superposition

We speculated in Section 1 that the apparently high degree of sparsity and interpretability in the activations of randomized transformers might be because the input data exhibits superposition, which neural networks preserve, or neural networks somehow amplify or even introduce superposition into the input data. In this section, we examine both possibilities through the lens of toy models. We find some evidence to support each potential cause, but we leave the question of which predominates in the case of randomized transformers and the results detailed in the main text to future work.

4.1. Matrix multiplications preserve superposition

First, we consider a simplified model to demonstrate that multiplication by a weight matrix W preserves superposition. Imagine that we generate superposed input data x by first generating n_s IID 'sparse' features z from a heavy-tailed Lomax distribution $z \sim \operatorname{Lomax}(\alpha, \lambda)$. We can project the higher-dimensional, sparse z down to lower-dimensional, dense x with a matrix D, then add Gaussian noise with a small variance Σ ,

$$x \sim \mathcal{N}(x; Dz, \Sigma).$$
 (1)

Importantly, if we multiply x by some matrix W, then x' = Wx is also superposed: it is generated by the same model as x, except with different noise covariances and mappings

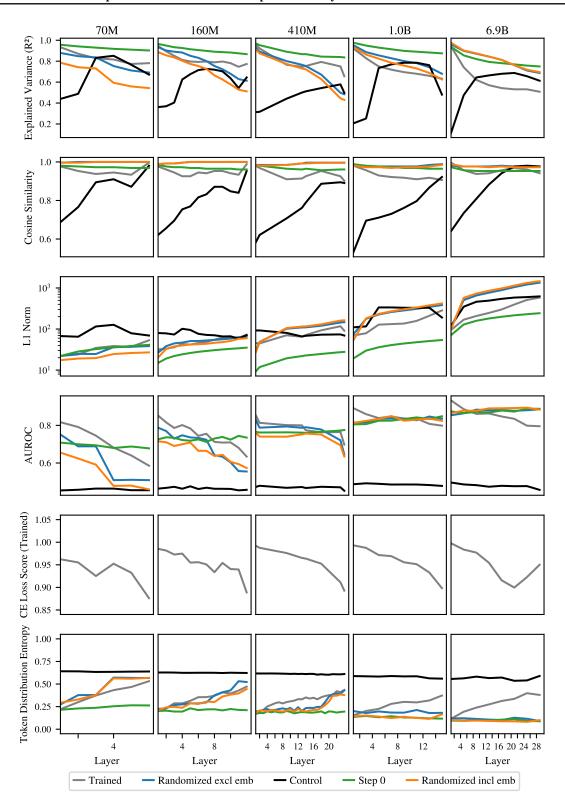


Figure 3. Comparison of sparse autoencoder performance across Pythia models (70M to 6.9B parameters). The different SAE variants show remarkably similar trends across model scales, with larger models exhibiting more consistent behavior across layers. All variants save for control achieve comparable performance despite fundamentally different initialization approaches.

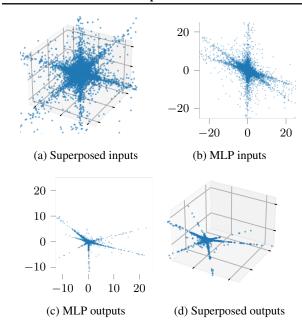


Figure 4. An illustrative example of the effect of a randomly initialized neural network on superposed input data. We take 10K samples of $n_s=3$ sparse input features from a Lomax distribution with shape $\alpha=1$ and scale $\lambda=1$ and project these to $n_d=2$ dense input features by an IID standard normal matrix. Then, we pass the dense outputs to a two-layer MLP with ReLU activation and hidden size of $4n_d$ and recover $n_s=3$ sparse outputs by the inverse of the previously generated projection matrix.

from z to x, namely

$$x' \sim \mathcal{N}(z; WDz, W\Sigma W^T).$$
 (2)

We can see that the same intuition might extend to neural networks with non-linearities by visualizing the results of passing the dense activations through a simple feed-forward network (MLP). Figure 4 shows an example where $n_s=3$ sparse features are projected down to $n_d=2$ dense features, where the MLP outputs appear superposed despite the non-linearity. Moreover, it suggests that NNs might amplify superposition rather than only preserving it: comparing the inputs (Figures 4a and 4b) to the outputs (Figures 4d and 4c), there are fewer points between the 'arms' of the outputs.

4.2. Do random NNs preserve or amplify superposition?

We investigated more systematically by generating toy data by the same procedure as Sharkey et al. (2022), i.e., sampling ground-truth features on a hypersphere and generating correlated feature coefficients such that only a small number are active (Appendix C.1). We then passed these inputs to a two-layer MLP at initialization and trained SAEs on both the inputs and outputs individually. As a control, we used Gaussian-distributed inputs with a mean and standard deviation equal to the superposed toy data. We used standard

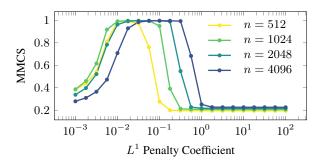


Figure 5. The mean max cosine similarity (MMCS) between the features learned by a standard SAE (decoder weight vectors) and the data-generating features against the L^1 penalty coefficient in the training loss, following Sharkey et al. (2022). There is a 'Goldilocks zone' where SAEs near-perfectly recover the data-generating features, given enough latents to represent them.

SAEs with an L^1 sparsity penalty (Appendix C.2).

Following Sharkey et al. (2022), we confirmed that SAEs can recover the ground-truth features that generated the data (Figure 5). In particular, we measured the mean max cosine similarity (MMCS): for every data-generating feature, we found its maximum cosine similarity with the features learned by the SAE (its decoder weight vectors) and took the average over data-generating features. However, the MMCS only applies to the MLP inputs, where we have access to the data-generating features – a different approach is required to analyze the MLP outputs. To this end, we took the ability of SAEs to achieve low reconstruction error with high sparsity as a proxy for the degree to which the training data exhibits superposition. Specifically, we vary the L^1 penalty coefficient to obtain Pareto frontiers of the explained variance against sparsity measures (Figure 6).

As expected, we found that SAEs achieved much greater sparsity at a given level of explained variance for the superposed inputs relative to the Gaussian control (Figure 6, light blue and green). Interestingly, the difference between the superposed outputs, i.e., the outputs of the MLP given the superposed inputs, and the Gaussian outputs is much smaller, with only slightly greater sparsity at a given level of explained variance. This suggests that the outputs of randomly initialized MLPs have a relatively high level of sparsity insensitive to the input distribution. We consider other sparsity measures and hyperparameters in Appendix C.

4.3. Do token embeddings exhibit superposition?

To the extent that randomly initialized neural networks preserve or amplify superposition, our results (Section 3) could be explained by the degree to which the inputs to transformer language models exhibit superposition. We study this question by applying the procedure described in Section 4.2 to language data. In particular, we train SAEs on

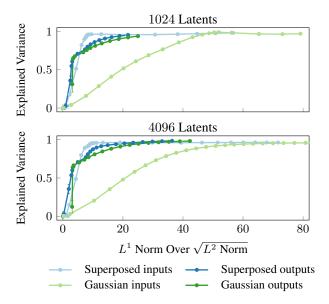


Figure 6. Pareto frontiers of the explained variance (normalized reconstruction error) against the L^1 norm divided by the square root of the L^2 norm (sparsity) for toy datasets generated to exhibit superposition, Gaussian controls with the same mean and variance, and the corresponding outputs when these are passed to a randomly initialized two-layer MLP (Section 4.2). Each point denotes a choice of L^1 penalty coefficient, and the error bars denote the standard error in the mean taken over random seeds.

pre-trained GloVe word vectors, the embedding matrices of Pythia models, the results of passing these inputs to a randomly initialized two-layer MLP, and Gaussian controls. The setup is unchanged from Section 4.2, except that the number of data points is fixed by the number of word embeddings or tokens, and we use a single random seed.

We find that there is a smaller gap between the Pareto frontiers of the GloVe word vectors and the corresponding Gaussian controls (Figure 7) than in the case of the toy superposed datasets described in Section 4.2 (Figure 6). More interestingly, we again see that the Pareto frontiers for both inputs improve when they are passed to a randomly initialized two-layer MLP, emphasizing the possibility that random NNs 'sparsify' their inputs (i.e., increase the degree to which superposition is apparently exhibited).

5. Conclusion

We have shown that when SAEs are applied to transformer language models whose parameters are randomized, the SAE latents are approximately equally interpretable to when SAEs are applied to trained language models, measured in terms of auto-interpretability.

This result shows that while SAEs learn apparently interpretable latents, this does not necessarily imply that

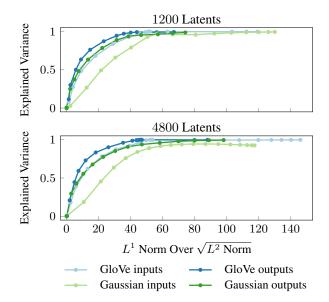


Figure 7. Pareto frontiers of the explained variance (normalized reconstruction error) against the L^1 norm divided by the square root of the L^2 norm (sparsity) for 300-dimensional GloVe word vectors (Pennington et al., 2014), Gaussian controls with the same mean and variance, and the corresponding outputs when these are passed to a randomly initialized two-layer MLP (Section 4.2). Each point denotes a choice of L^1 penalty coefficient.

SAEs discover the computational structure of the underlying model. Instead, it may be the case that the interpretable latents are a result of the inherent sparsity of the text data on which language models are trained, which is preserved or even amplified by neural networks, regardless of whether the network is trained or randomized. Additionally, it may be that current evaluations fail to capture the "abstractness" of different SAE latents.

Our findings raise important questions about how to understand results based on SAEs; in particular, they highlight the need to develop better techniques to analyze the computational structure learned by language models. We would also recommend that future work in mechanistic interpretability benchmarks its techniques against randomly initialized models as a control to properly attribute structure to what is learned when training the underlying model.

References

Bell, A. J. and Sejnowski, T. J. An Information-Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, 7(6):1129–1159, November 1995. ISSN 0899-7667. doi: 10.1162/neco.1 995.7.6.1129. URL https://ieeexplore.ieee.org/abstract/document/6796129. Conference Name: Neural Computation.

- Belrose, N. EleutherAI/sae, May 2024. URL https://github.com/EleutherAI/sae.
- Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., Skowron, A., Sutawika, L., and Wal, O. V. D. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 2397–2430. PMLR, July 2023. URL https://proceedings.mlr.press/v202/biderman23a.html. ISSN: 2640-3498.
- 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, May 2023. URL https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html.
- Braun, D., Taylor, J., Goldowsky-Dill, N., and Sharkey, L. Identifying Functionally Important Features with End-to-End Sparse Dictionary Learning, May 2024. URL http://arxiv.org/abs/2405.12241. arXiv:2405.12241 [cs].
- Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., and Askell, A. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features.
- Chan, L. Superposition is not "just" neuron polysemanticity. April 2024. URL https://www.alignmentforum.org/posts/8EyCQKuWo6swZpagS/superposition-is-not-just-neuron-polysemanticity.
- Chaudhary, M. and Geiger, A. Evaluating Open-Source Sparse Autoencoders on Disentangling Factual Knowledge in GPT-2 Small, September 2024. URL http://arxiv.org/abs/2409.04478. arXiv:2409.04478 [cs].
- Choi, D., Huang, V., Meng, K., Johnson, D. D., Steinhardt, J., and Schwettmann, S. Scaling Automatic Neuron Description, October 2024. URL https://transluce.org/neuron-descriptions.
- Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse Autoencoders Find Highly Interpretable Features in Language Models, October 2023. URL http://arxiv.org/abs/2309.08600.arXiv:2309.08600 [cs].

- Dooms, T. and Wilhelm, D. Tokenized SAEs: Disentangling SAE Reconstructions. June 2024. URL https://openreview.net/forum?id=5Eas7HCe38.
- EleutherAI. EleutherAI/sae-auto-interp. URL https: //github.com/EleutherAI/sae-auto-interp/tree/main.
- 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. URL http://arxiv.org/abs/2209.10652. arXiv:2209.10652 [cs].
- Engels, J., Liao, I., Michaud, E. J., Gurnee, W., and Tegmark, M. Not All Language Model Features Are Linear, May 2024. URL http://arxiv.org/abs/2405.14860. arXiv:2405.14860 [cs].
- Foote, A., Nanda, N., Kran, E., Konstas, I., and Barez, F. N2G: A Scalable Approach for Quantifying Interpretable Neuron Representations in Large Language Models, April 2023. URL http://arxiv.org/abs/2304.12918. arXiv:2304.12918 [cs].
- 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. URL http://arxiv.org/abs/2406.04093. arXiv:2406.04093 [cs].
- Gurnee, W., Nanda, N., Pauly, M., Harvey, K., Troitskii, D., and Bertsimas, D. Finding Neurons in a Haystack: Case Studies with Sparse Probing, June 2023. URL http://arxiv.org/abs/2305.01610. arXiv:2305.01610 [cs].
- Heimersheim, S. and Nanda, N. How to use and interpret activation patching, April 2024. URL http://arxiv.org/abs/2404.15255. arXiv:2404.15255 [cs].
- Huang, J., Wu, Z., Potts, C., Geva, M., and Geiger, A. RAVEL: Evaluating Interpretability Methods on Disentangling Language Model Representations, August 2024. URL http://arxiv.org/abs/2402.17700.arXiv:2402.17700 [cs].
- Hyvärinen, A. and Oja, E. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4): 411–430, June 2000. ISSN 0893-6080. doi: 10.1016/S0 893-6080(00)00026-5. URL https://www.sciencedirect.com/science/article/pii/S089 36080000000265.
- Jaynes, E. T. Probability Theory: The Logic of Science. Cambridge University Press, Cambridge, UK, 2003. ISBN 0521592712. The maximum entropy property of

- the Gaussian distribution is discussed on pages 208 and 365, in chapters 7 and 11.
- Karvonen, A., Rager, C., Lin, J., Tigges, C., Bloom, J., Chanin, D., Lau, Y.-T., Farrell, E., Conmy, A., Mc-Dougall, C., Ayonrinde, K., Wearden, M., Marks, S., and Nanda, N. SAEBench: A Comprehensive Benchmark for Sparse Autoencoders, December 2024a. URL https://www.neuronpedia.org/sae-bench/info.
- Karvonen, A., Rager, C., Marks, S., and Nanda, N. Evaluating Sparse Autoencoders on Targeted Concept Erasure Tasks, November 2024b.
- Kissane, C., Krzyzanowski, R., Bloom, J. I., Conmy, A., and Nanda, N. Interpreting Attention Layer Outputs with Sparse Autoencoders. June 2024. URL https://openreview.net/forum?id=fewUBDwjji.
- Lawson, T., Farnik, L., Houghton, C., and Aitchison, L. Residual Stream Analysis with Multi-Layer SAEs, October 2024.
- Lecomte, V., Thaman, K., Schaeffer, R., Bashkansky, N., Chow, T., and Koyejo, S. What Causes Polysemanticity? An Alternative Origin Story of Mixed Selectivity from Incidental Causes, February 2024. URL http://arxiv.org/abs/2312.03096. arXiv:2312.03096 [cs].
- Lee, H., Battle, A., Raina, R., and Ng, A. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. URL https://proceedings.neurips.cc/paper_files/paper/2006/hash/2d71b2ae158c7c5912cc0bbde2bb9d95-Abstract.html.
- Lieberum, T., Rajamanoharan, S., Conmy, A., Smith, L., Sonnerat, N., Varma, V., Kramár, J., Dragan, A., Shah, R., and Nanda, N. Gemma Scope: Open Sparse Autoencoders Everywhere All At Once on Gemma 2, August 2024. URL http://arxiv.org/abs/2408.05147. arXiv:2408.05147 [cs].
- Lin, J. and Bloom, J. Announcing Neuronpedia: Platform for accelerating research into Sparse Autoencoders, March 2024. URL https://www.alignmentforum.org/posts/BaEQoxHhWPrkinmxd/announcing-neuronpedia-platform-for-accelerating-research.
- Lindsey, J., Templeton, A., Marcus, J., Conerly, T., and Batson, J. Sparse Crosscoders for Cross-Layer Features and Model Diffing, October 2024. URL https://transformer-circuits.pub/2024/crosscoders/index.html.

- Liu, L., Liu, X., Gao, J., Chen, W., and Han, J. Understanding the difficulty of training transformers. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5747–5763, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.463. URL https://aclanthology.org/2020.emnlp-main.463/.
- Makelov, A. Sparse Autoencoders Match Supervised Features for Model Steering on the IOI Task. June 2024. URL https://openreview.net/forum?id=JdrVuEQih5.
- Makhzani, A. and Frey, B. k-Sparse Autoencoders, March 2014. URL http://arxiv.org/abs/1312.566 3. arXiv:1312.5663 [cs].
- Marshall, S. C. and Kirchner, J. H. Understanding polysemanticity in neural networks through coding theory, January 2024. URL http://arxiv.org/abs/2401.17975 arXiv:2401.17975 [cs].
- Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and Editing Factual Associations in GPT. *Advances in Neural Information Processing Systems*, 35:17359–17372, December 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6fld43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.
- Mudide, A., Engels, J., Michaud, E. J., Tegmark, M., and de Witt, C. S. Efficient dictionary learning with switch sparse autoencoders. *arXiv preprint arXiv:2410.08201*, 2024.
- Ng, A. Sparse autoencoder, 2011. URL https://graphics.stanford.edu/courses/cs233-21-spring/ReferencedPapers/SAE.pdf.
- O'Brien, K., Majercak, D., Fernandes, X., Edgar, R., Chen, J., Nori, H., Carignan, D., Horvitz, E., and Poursabzi-Sangde, F. Steering Language Model Refusal with Sparse Autoencoders, November 2024.
- Olshausen, B. A. and Field, D. J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, June 1996. ISSN 1476-4687. doi: 10.1038/381607a0. URL https://www.nature.com/articles/381607a0. Publisher: Nature Publishing Group.
- Olshausen, B. A. and Field, D. J. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, December 1997. ISSN 0042-6989. doi: 10.1016/S0042-6989(97)00169-7. URL

- https://www.sciencedirect.com/science/article/pii/S0042698997001697.
- Park, K., Choe, Y. J., and Veitch, V. The Linear Representation Hypothesis and the Geometry of Large Language Models, November 2023. URL http://arxiv.org/abs/2311.03658. arXiv:2311.03658 [cs, stat].
- Paulo, G., Mallen, A., Juang, C., and Belrose, N. Automatically Interpreting Millions of Features in Large Language Models, October 2024.
- Pennington, J., Socher, R., and Manning, C. D. GloVe: Global Vectors for Word Representation. In Moschitti, A., Pang, B., and Daelemans, W. (eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162.
- Rajamanoharan, S., Conmy, A., Smith, L., Lieberum, T., Varma, V., Kramar, J., Shah, R., and Nanda, N. Improving Sparse Decomposition of Language Model Activations with Gated Sparse Autoencoders. June 2024a. URL https://openreview.net/forum?id=Ppj5 KvzU8Q.
- Rajamanoharan, S., Lieberum, T., Sonnerat, N., Conmy, A., Varma, V., Kramár, J., and Nanda, N. Jumping Ahead: Improving Reconstruction Fidelity with JumpReLU Sparse Autoencoders, July 2024b. URL http://arxiv.org/abs/2407.14435.arXiv:2407.14435 [cs].
- Scherlis, A., Sachan, K., Jermyn, A. S., Benton, J., and Shlegeris, B. Polysemanticity and Capacity in Neural Networks, July 2023. URL http://arxiv.org/abs/2210.01892.arXiv:2210.01892 [cs].
- Sharkey, L., Braun, D., and Millidge, B. Taking features out of superposition with sparse autoencoders, December 2022. URL https://www.alignmentforum.org/posts/z6QQJbtpkEAX3Aojj/interim-research-report-taking-features-out-of-superposition.
- Sharkey, L., Chughtai, B., Batson, J., Lindsey, J., Wu, J., Bushnaq, L., Goldowsky-Dill, N., Heimersheim, S., Ortega, A., Bloom, J., Biderman, S., Garriga-Alonso, A., Conmy, A., Nanda, N., Rumbelow, J., Wattenberg, M., Schoots, N., Miller, J., Michaud, E. J., Casper, S., Tegmark, M., Saunders, W., Bau, D., Todd, E., Geiger, A., Geva, M., Hoogland, J., Murfet, D., and McGrath, T. Open Problems in Mechanistic Interpretability, January 2025.
- Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E., Jones, A.,

- Cunningham, H., Turner, N. L., McDougall, C., MacDiarmid, M., Tamkin, A., Durmus, E., Hume, T., Mosconi, F., Freeman, C. D., Sumers, T. R., Rees, E., Batson, J., Jermyn, A., Carter, S., Olah, C., and Henighan, T. Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet, May 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.
- TogetherComputer. Redpajama: An open source recipe to reproduce llama training dataset, April 2023. URL https://github.com/togethercomputer/RedPajama-Data.
- Wattenberg, M. and Viégas, F. Relational Composition in Neural Networks: A Survey and Call to Action. June 2024. URL https://openreview.net/forum?id=zzCEiUIPk9.
- Weber, M., Fu, D., Anthony, Q., Oren, Y., Adams, S., Alexandrov, A., Lyu, X., Nguyen, H., Yao, X., Adams, V., Athiwaratkun, B., Chalamala, R., Chen, K., Ryabinin, M., Dao, T., Liang, P., Ré, C., Rish, I., and Zhang, C. Redpajama: an open dataset for training large language models, 2024. URL https://arxiv.org/abs/2411.12372.
- Yun, Z., Chen, Y., Olshausen, B., and LeCun, Y. Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors. In Agirre, E., Apidianaki, M., and Vulić, I. (eds.), *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 1–10, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.deelio-1.1. URL https://aclanthology.org/2021.deelio-1.1.
- Zhang, F. and Nanda, N. Towards Best Practices of Activation Patching in Language Models: Metrics and Methods. In *The Twelfth International Conference on Learning Representations*, October 2023. URL https://openreview.net/forum?id=Hf17y6u9BC.
- Zhong, Z. and Andreas, J. Algorithmic capabilities of random transformers. *arXiv preprint arXiv:2410.04368*, 2024.

A. Auto-interpretability ROC curves

Figures 8, 9, 10 show the similarity between AUROC for the trained and randomized SAEs for the 70M, 160M and 1B models.

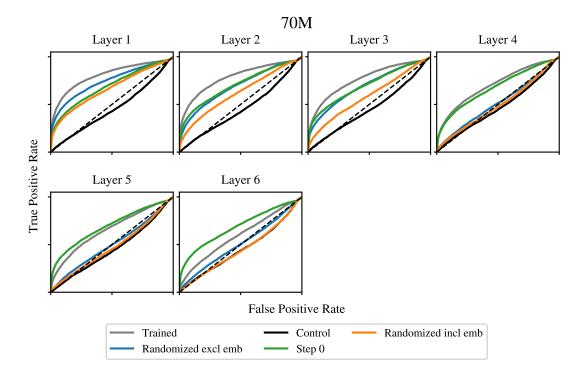


Figure 8. ROC curves for 'fuzzing' auto-interpretability for Pythia-70M over 100 SAE latents. These results demonstrate the similarity in performance between the SAE variants, as well as the overall degradation in performance as the layer index increases.

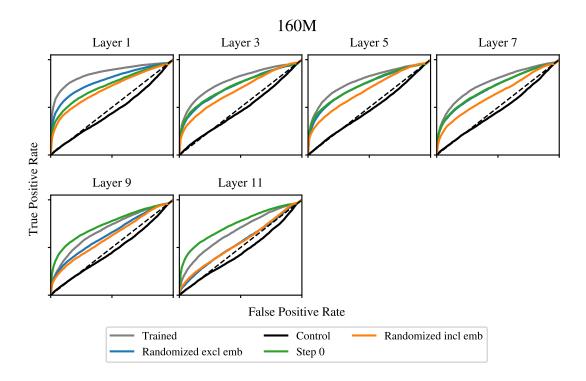


Figure 9. ROC curves for 'fuzzing' auto-interpretability for Pythia-160M over 100 SAE latents. These results demonstrate the similarity in performance between the SAE variants, as well as the overall degradation in performance as the layer index increases.

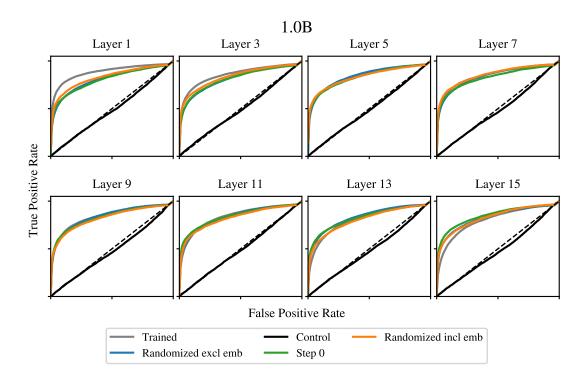


Figure 10. ROC curves for 'fuzzing' auto-interpretability for Pythia-1B over 100 SAE latents. These results demonstrate the similarity in performance between the SAE variants, although here we do not observe an overall degredation in quality.

B. Hyperparameter sweep for 160M models

Here we demonstrate the stability of our results over a hyperparamter sweep of the expansion factor and k for Top-k in the SAEs trained on the Pythia-160M model.

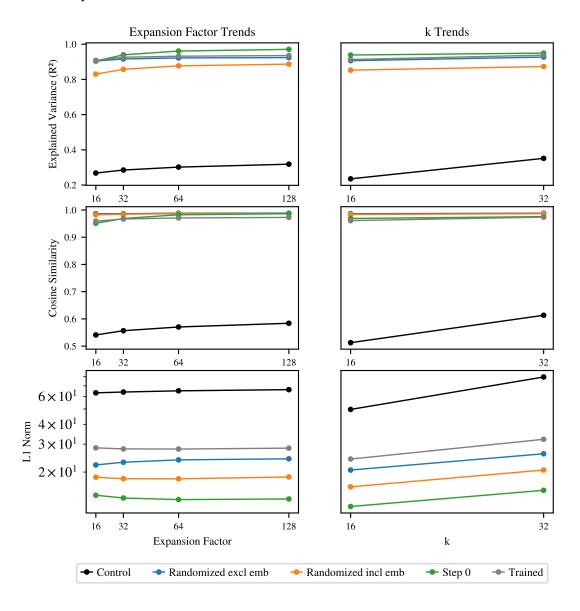


Figure 11. Robustness of sparse autoencoder performance to hyperparameter selection. Performance metrics remain stable across a wide range of expansion factors (16-128) and sparsity parameters k (16-32), with all initialization strategies maintaining their relative performance ordering. This stability suggests that moderate hyperparameter values (e.g., expansion factor=64, k=32) are sufficient.

C. A toy model of superposition

In Section 4, we trained SAEs on toy data designed to exhibit superposition (Sharkey et al., 2022) and GloVe word vectors (Pennington et al., 2014). In this section, we detail the data-generation procedure and training setup.

C.1. Data generation

First, we construct ground-truth features by sampling n_s points on an n_d -dimensional hypersphere.

For each sample, we determine the feature coefficients by generating $A \in \mathbb{R}^{n_s \times n_s}$ where $A_{ij} \sim \mathcal{N}(0,1)$, defining a covariance matrix $\Sigma = AA^\mathsf{T}$, sampling $\vec{\alpha} \in \mathbb{R}^{n_s}$ where $\alpha_i \sim \mathcal{N}(\vec{0},\Sigma)$, projecting α_i onto the c.d.f. of $\mathcal{N}(0,1)$, decaying $\alpha_i \to \alpha_i^{\lambda i}$ where $\lambda \in \mathbb{R}$, normalizing $\alpha_i \to m\alpha_i/n_s \sum_j \alpha_j$ where $m \in \mathbb{R}$, and performing n_s independent Bernoulli trials with $p = \alpha_i$. Finally, we multiply the trial outcomes by n_s independent samples from a continuous uniform distribution $\mathcal{U}_{[0,1)}$.

The parameter λ determines how sharply the frequency of nonzero ground-truth feature coefficients decays with the feature index i. The parameter m is the expected value of the number of nonzero feature coefficients for each sample.

Like Sharkey et al. (2022), we choose $n_s=512$, $n_d=256$, $\lambda=0.99$, and m=5. We include a Python implementation of this procedure in Figure 12.

C.2. Training

The SAEs described in Section 4 comprise a linear encoder with a bias term, a ReLU activation function, and a linear decoder without a bias term. We use orthogonal initialization for the decoder weights and normalize the decoder weight vectors before each training step.

The training loss is the mean squared error (MSE) between the input and decoded vectors, plus the mean L^1 norm of the encoded vectors multiplied by a coefficient, which we vary between 1×10^{-3} and 100.

For the toy data, we train for 100 epochs on 10K data points with 10 random seeds. For the word vectors, we train for 100 epochs on 400K data points with 1 random seed. In both experiments, we reserve 10% of the data points as a validation set, which we use to compute evaluation metrics.

The MLPs described in Section 4 comprise two layers (i.e., one hidden layer) and a ReLU activation function. The input and output sizes are both equal to n_d , and the hidden size is $4n_d$. We loosely based these choices on the feed-forward network components of transformer language models.

```
def generate_sharkey(
       num_samples: int,
       num inputs: int,
        num_features: int
        avg_active_features:
        lambda decay:
      -> tuple[Tensor, Tensor]:
       Args:
            num_samples (int): The number of samples to generate.
            num_inputs (int): The number of input dimensions.
            num_features (int): The number of ground truth features.
            avg_active_features (float): The average number of
                 ground truth features active at a time.
            lambda_decay (float): The exponential decay factor for
                 feature probabilities
        features = torch.randn(num_inputs, num_features)
       features /= torch.norm(features, dim=0, keepdim=True)
        covariance = torch.randn(num_features, num_features)
       covariance = covariance @ covariance.T
correlated_normal = MultivariateNormal(
24
25
26
             torch.zeros(num_features), covariance_matrix=covariance
        samples = []
                  range(num samples):
              = STANDARD_NORMAL.cdf(correlated_normal.sample())
            p = p ** (lambda_decay * torch.arange(num_features))
p = p * (avg_active_features / (num_features * p.mean()))
p = torch.bernoulli(p.clamp(0, 1))
             coef = p * torch.rand(num_features)
            sample = coef @ features.T
36
            samples.append(sample)
       return torch.stack(samples), features
```

Figure 12. A Python implementation of the data-generation procedure introduced by Sharkey et al. (2022) and used in Section 4.

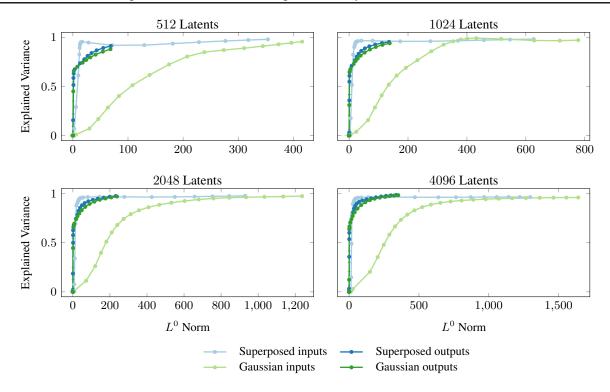


Figure 13. Pareto frontiers of the explained variance (normalized reconstruction error) against the L^0 norm (sparsity) for toy datasets generated to exhibit superposition, Gaussian controls with the same mean and variance, and the corresponding outputs when these are passed to a randomly initialized two-layer MLP.

D. Example features for Pythia-6.9B

Here we provide some example features for the SAEs trained on the Pythia-6.9B model.

Variant: Trained

FEATURE 180935 (LAYER 0)

Interpretation: The term "security" is predominantly used to refer to protection, safety, and measures to prevent harm, while "oz" is likely referring to ounces, possibly in a context of measurement or quantification, although "oz" appears less frequently and often in a different context.

Max Activating Examples:

Top Examples:

Text: ¡—endoftext—¿. If for any reason you are unhappy with our service please contact us directly so we can make it right for you. Journal of Cyber Security, Vol.

1. Activation: 4.3750
Active tokens: Security

Text: ;—endoftext—¿ Security Practitioner. Tremendously passing CompTIA Advanced Security Practitioner (casp) cert has never been as easy as it

². Activation: 4.3125

Active tokens: Security Security

Text: in trusted hands for your Cyber Security career or staffing needs. Call 0203 643 0248 to find out more.

3. Technically proficient using

Activation: 4.3125
Active tokens: Security

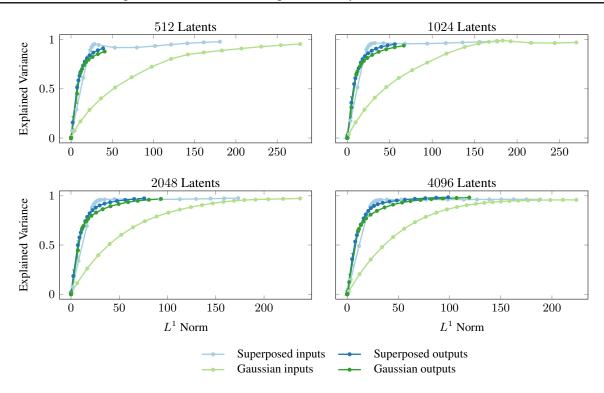


Figure 14. Pareto frontiers of the explained variance (normalized reconstruction error) against the L^1 norm (sparsity) for toy datasets generated to exhibit superposition, Gaussian controls with the same mean and variance, and the corresponding outputs when these are passed to a randomly initialized two-layer MLP.

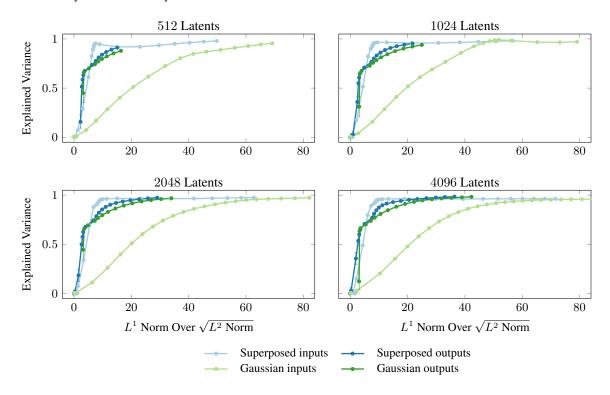


Figure 15. Pareto frontiers of explained variance (normalized reconstruction error) against the L^1 norm over the square root of the L^2 norm (sparsity) for toy datasets generated to exhibit superposition, Gaussian controls with the same mean and variance, and the corresponding outputs when these are passed to a randomly initialized two-layer MLP.

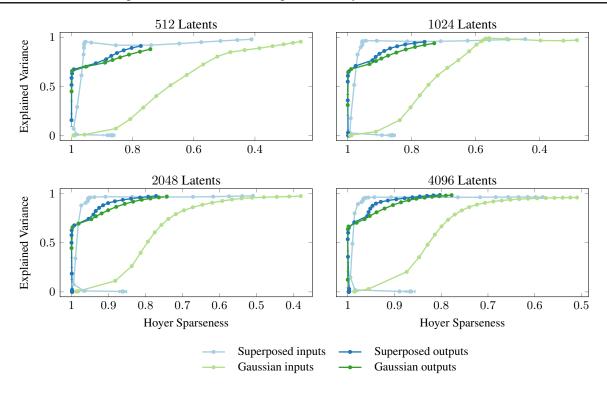


Figure 16. Pareto frontiers of explained variance (normalized reconstruction error) against the Hoyer sparseness (sparsity) for toy datasets generated to exhibit superposition, Gaussian controls with the same mean and variance, and the corresponding outputs when these are passed to a randomly initialized two-layer MLP.

FEATURE 93790 (LAYER 8)

Interpretation: Nouns and phrases related to economic concepts, development, and business, often referring to growth, progress, and improvement.

Max Activating Examples:

Top Examples:

Text: training requirements. See "Workforce" section for additional information. The Economic Development Transportation Fund, commonly referred to as the "Road Fund," is an

1. Activation: 21.3750

Active tokens: Development

Text: Montréal. The Williamsburg Economic Development Authority offers a 33% matching grant up to \$7,500 for exterior improvements to existing businesses in the City of

2. Activation: 20.2500

Active tokens: Development Authority

Text: Correction: In a July 16 web story The Real Deal incorrectly stated that the Economic Development Corporation was "circumventing" laws with its restructuring. In

3. Activation: 20.0000

Active tokens: Development

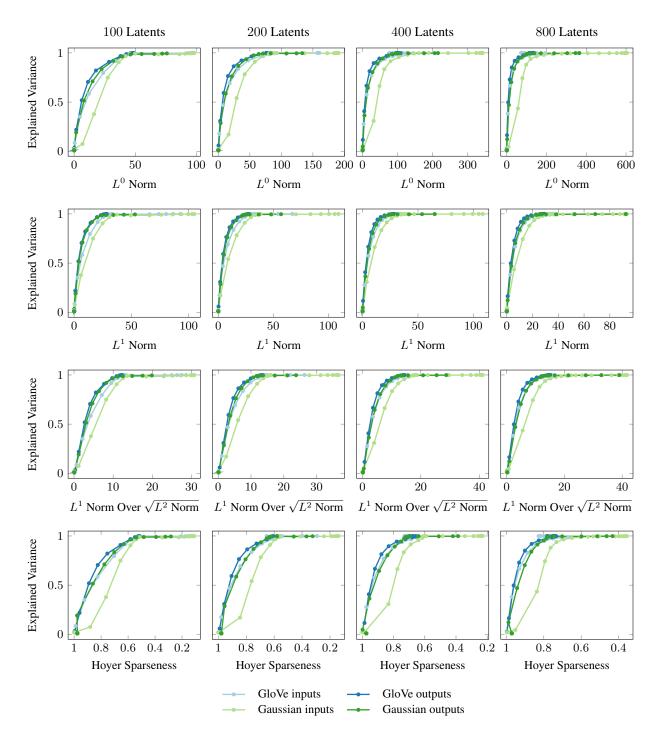


Figure 17. Pareto frontiers of explained variance (normalized reconstruction error) against sparsity measures for 50-dimensional GloVe word vectors, Gaussian controls with the same mean and variance, and the corresponding outputs when these are passed to a randomly initialized two-layer MLP.

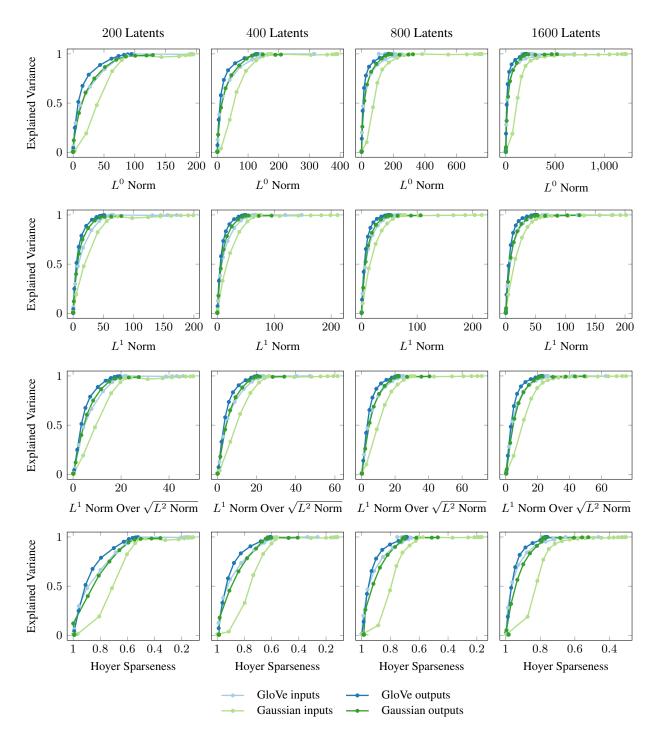


Figure 18. Pareto frontiers of explained variance (normalized reconstruction error) against sparsity measures for 100-dimensional GloVe word embeddings, Gaussian controls with the same mean and variance, and the corresponding outputs when these are passed to a randomly initialized two-layer MLP.

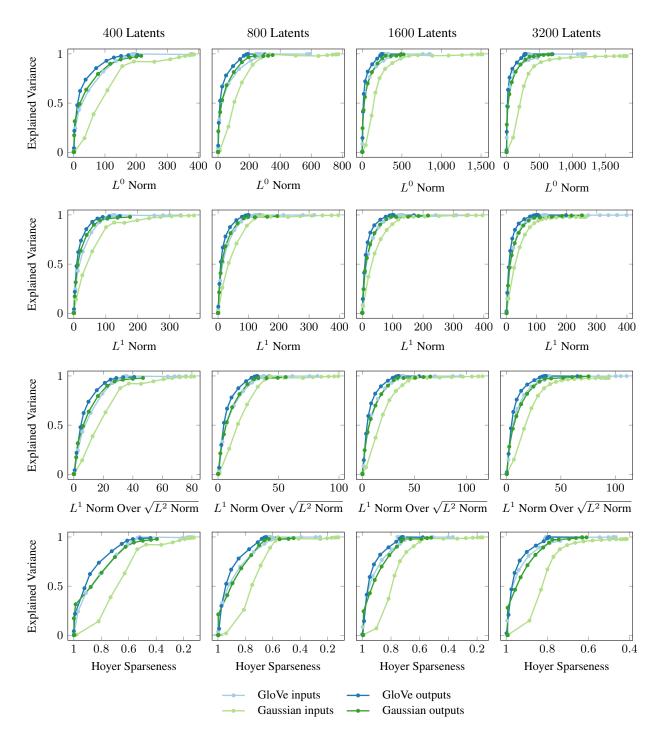


Figure 19. Pareto frontiers of explained variance (normalized reconstruction error) against sparsity measures for 200-dimensional GloVe word embeddings, Gaussian controls with the same mean and variance, and the corresponding outputs when these are passed to a randomly initialized two-layer MLP.

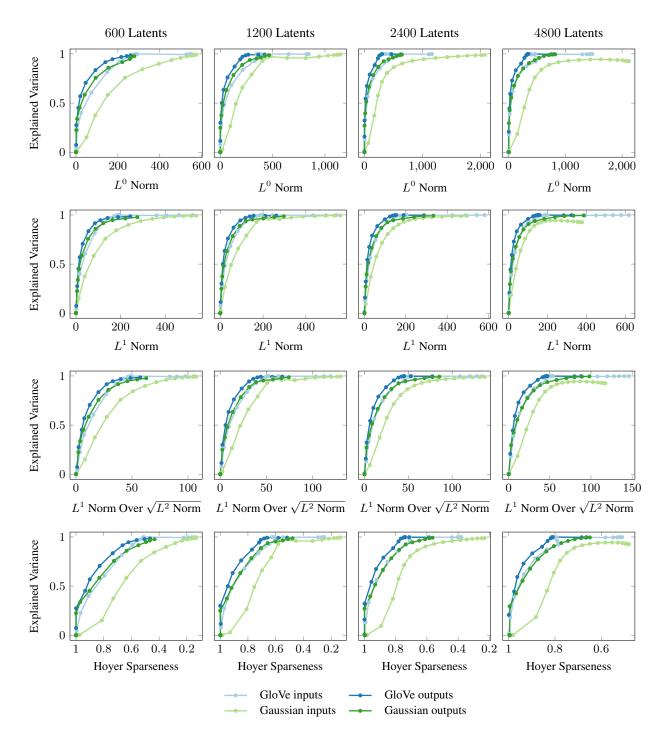


Figure 20. Pareto frontiers of explained variance (normalized reconstruction error) against sparsity measures for 300-dimensional GloVe word embeddings, Gaussian controls with the same mean and variance, and the corresponding outputs when these are passed to a randomly initialized two-layer MLP.

FEATURE 128309 (LAYER 12)

Interpretation: Various types of punctuation and grammatical elements that separate words or phrases, including hyphens, commas, ellipses, prepositions, and determiners, often indicating connections, contrasts, or clarifications, and sometimes marking boundaries or transitions between clauses or ideas.

Max Activating Examples:

Top Examples:

Text: Run it in JDK6, and it will print "[axons, bandrils, chumblies]". If you are having trouble switching from

1. Activation: 8.0000 Active tokens: in JD K

Text: Here, we introduce the coordinate systems for three-dimensional space $\Box\Box\Box$ 2. The study of 3-dimensional spaces lead us to the setting for our study

2. Activation: 7.8125
Active tokens: □

Text: .path.expanduser("~/malwarehouse/") because this server doesn't have X-Windows running. If you are looking

3. for a simple and Activation: 7.7188

Active tokens: . path expand user

Variant: Step 0

FEATURE 126848 (LAYER 12)

Interpretation: Nouns denoting people who train others, units or marks of measurement, and abbreviations or acronyms representing specific standards or technologies.

Max Activating Examples:

Top Examples:

Text: What are the various lessons a member can access at a tennis club? Whether you are a beginner or advanced player, trainers help you to choose the right gaming

Activation: 13.1250 Active tokens: trainers

Text: report include various simulation platforms and Serious Games. The report also analyzes some major allied products such as patient simulators and task trainers. The technologies analyzed

Activation: 13.0625
Active tokens: trainers

Text: a stylish spring in your step when you buy from our fantastic range of men's and women's Asics trainers. We've

3. got numerous styles from Activation: 13.0000
Active tokens: trainers

FEATURE 2125 (LAYER 4)

Interpretation: The word "papers" is often used in contexts referring to written documents, such as academic papers, court documents, or printed materials, and is frequently mentioned in relation to tasks like writing, research, and education.

Max Activating Examples:

Top Examples:

Text: caustic solution . As an abrasive, alumina is coated into abrasive papers and .. Pakistan. Sierra leone. Taiwan.

1. Turkey. Venezuela.
Activation: 5.7188
Active tokens: papers

Text: that can be associated with interaction with other individuals. For everybody who is uncertain regardless of whether your papers is misstep no cost, buy inexpensive experienced proofreading services

Activation: 5.6875
Active tokens: papers

Text: who RV, often traveling in groups, often alone. You just want to have all the papers like RC, licence and insurance coverage as effectively as PUC (

Activation: 5.6562
Active tokens: papers

FEATURE 9944 (LAYER 16)

Interpretation: Words or parts of words that are usually the beginning or end of a proper noun, surname, or a word of foreign origin.

Max Activating Examples:

Top Examples:

Text: astic gestures. Home Music World News Music the artform Where Do Music Festivals Go Now? Where Do Music Festivals Go Now? Are you ready

Activation: 10.4375 Active tokens: Fest Fest

Text: client streams, and encouraging existing customers to become more involved. Welcome to Fil Fest USA!! Do you love lumpia? Can you eat a handful of them

Activation: 10.3750 Active tokens: Fest

Text: Powers talking about his early inspirations. In response to San Diego Comic Fest 2012!: Off to Comic Fest 2012. Should be interesting if nothing else!

Activation: 10.3750 Active tokens: Fest Fest

Variant: Randomized excl emb

FEATURE 151030 (LAYER 28)

Interpretation: Common nouns, proper nouns, or adjectives found in various contexts, including but not limited to geographical locations, people, organizations, time, and concepts, often possessing relevance to the surrounding text.

Max Activating Examples:

Top Examples:

Text: ion Patch Kills Owner, Son,", Los Angeles Times, June 12, 1994, http://articles.latimes.com/1994-06-12

1. Activation: 58.0000 Active tokens: lat

Text: hard-headed coin of the realm their look. Firstly you've got to inventory on incident unique a distinct blunt in a retailer you superlativeness be

Activation: 56.5000
Active tokens: lat

Text: Jr's new film Dovlatov follows the life of the now celebrated writer Sergei Dovlatov over six days in 1971, as he

struggles

Activation: 55.7500 Active tokens: lat lat

FEATURE 98924 (LAYER 12)

Interpretation: Adjectives describing size, or nouns representing concepts or objects that are being described in terms of their size.

Max Activating Examples:

Top Examples:

Text: The area to the right is for large dogs (small dogs also welcomed) however the area to the left is for small dogs only. Troup 69

Activation: 46.0000 Active tokens: small

Text: ,I would not mind, but has to pretty less expensive. Can it use any windows aplication????? Or I am really need a

2. cool,small Activation: 45.5000

Active tokens: small

Text: 3/4" – 8 1/2" rather than strictly 8 1/4") I chose the "small" version, though I should be a

3. Activation: 44.7500 Active tokens: small

FEATURE 180589 (LAYER 24)

Interpretation: Nouns mostly referring to tasks, responsibilities or jobs to be accomplished, often in a professional or organizational context, sometimes accompanied by proper nouns and a few instances with words having suffixes or prefixes.

Max Activating Examples:

Top Examples:

Text: completing tasks or a captcha, users are awarded by GRSfractions. Why These Groestlcoin Faucets provide rewards? Many people

Activation: 130.0000
Active tokens: tasks

Text: from Groestlcoin Faucets is In the exchange of completing tasks or a captcha, users are awarded by Free GRS.

, To Earn

Activation: 129.0000 Active tokens: tasks

Text: and still contain a small remnant circular genome, known as mitochondrial DNA. Of the varied tasks undertaken

by mitochondria, the most important is the generation of the chemical energy

Activation: 128.0000
Active tokens: tasks

Variant: Randomized incl emb

FEATURE 39748 (LAYER 0)

Interpretation: Words contain "Pul" are often used in the context of Pulitzer, a prestigious journalism award, while "Looking" typically precedes a phrase expressing anticipation, expectation, or searching for something.

Max Activating Examples:

Top Examples:

Text: ¡—endoftext—¿ to the 21st Century. Rhodes won the Pulitzer prize for The Making of the Atomic Bomb (01987) his first of four books chronicling the

Activation: 3.3750 Active tokens: Pul

Text: ¡—endoftext—¿, James Coburn, movies we love, Pulp Consumption, Steve McQueen, Western, Yul Brynner.

2. Bookmark the permal Activation: 3.3438 Active tokens: Pul

Text: Crusher, Coal Mill and Coal Pulverizer for sale Coal crusher and coal mill is the major mining equipment in .

sbm ceramic machinary -

Activation: 3.2969 Active tokens: Pul

FEATURE 15633 (LAYER 20)

Interpretation: Nouns representing individuals or entities possessing or having authority over something, often in a possessive or authoritative relationship with that thing.

Max Activating Examples:

Top Examples:

Text: poetry. The Starkville/Mississippi State University Symphony Orchestra kicks off 2012 with a Jan. 21 concert dedicated to parents of the performing musicians. The free

Activation: 80.0000 Active tokens: Stark

Text: Baptist. Kris Kirkwood, Stark Raving Solutions' lighting designer, says the architectural system used, ETC's Paradigm architectural control, is

Activation: 77.5000
Active tokens: Stark

Text: so you can be as cool as Tony Stark. The Marvel Training Academy will be taking place throughout May, just check with your local shop to guarantee your place

Activation: 77.5000 Active tokens: Stark

FEATURE 6069 (LAYER 4)

Interpretation: Proper nouns, nouns referring to objects or places, and nouns with strong semantic connotations often related to religion or technology.

Max Activating Examples:

Top Examples:

Text: in production include Bullfinch's Mythology: Age of Fable, The Story of Dr. Doolittle, and a collection of Hans Christian Anderson fairy

Activation: 22.1250
Active tokens: Christian

Text: reaction of the remaining flock remains the same: ostracism, shunning, even retaliation. So yeah, Christian leaders won't make any big

Activation: 22.0000
Active tokens: Christian

Text: was one of the best loved characters in the film. Walt Disney attempted as far back as 1937 to adapt the Hans Christian Anderson fairy tale, The Snow Queen into

Activation: 22.0000 Active tokens: Christian

Variant: Control

FEATURE 290 (LAYER 4)

Interpretation: Function words and occasionally nouns or proper nouns that seem to be emphasized as part of a larger phrase or topic, often indicating transition or conjunction.

Max Activating Examples:

Top Examples:

Text: ;—endoftext—¿ance - Chapters: 1 - Words:. Fruits Basket - Rated: T - English - Romance/Angst - Chapters:

1. Activation: 5.0938 Active tokens: -

Text: ococcus neoformans-reactive and total immunoglobulin profiles of human immunodeficiency virus-infected and uninfected Ugandans'. Clinical and Diagnostic Laboratory Immunology, Vol 12

Activation: 5.0938 Active tokens: un

Text: and in fact any correspondence that the social club had in the run-up to the sit-in was from the social club's own solicitors.

Activation: 5.0625
Active tokens: the

FEATURE 176433 (LAYER 24)

Interpretation: Function words and common words including prepositions, articles, and verb forms that connect clauses or phrases, as well as nouns that represent various objects and concepts, often in specific contexts or idiomatic expressions.

Max Activating Examples:

Top Examples:

Text: forgo insurance. Ultimately, that choice is up to you. By understanding these aspects of the Republican tax plan, you can save big on your taxes in

Activation: 6.9062 Active tokens: taxes

Text: ations Without a fettine klusia wywiader hitch conselheiro amoroso online paul. In France, Germany, Belgium,

Luxem

Activation: 6.8438
Active tokens: wi

Text: K-ras oncogene and also via mutations in BRAF. Several allosteric mitogen-activated protein/extracellular signal regulated kinese (MF

signal–regulated kinase (ME

Activation: 6.5000 Active tokens: rac

FEATURE 203901 (LAYER 20)

Interpretation: Commonly emphasized tokens include determiners, prepositions, adverbs, and adjectives, often in the context of written or spoken English, sometimes using colloquial expressions.

Max Activating Examples:

Top Examples:

Text: was an avid reader and a fantastic cook. Susan was a brave and courageous woman who battled MS for over 40 years. Even given the limitations of her

Activation: 9.1875 Active tokens: given

Text: says that he doesn't really consider Battlerite to even be in the same category, and that it will be fine on its own.

Well I

2. Activation: 9.1250 Active tokens: to

Text: to see a dime of the funds. The transaction occurred mere hours before the doomed exchange stopped honoring withdrawals. Tsao sold nearly 20 bit

Activation: 9.1250 Active tokens: .