Assessment Report

Summary

This project is a reproduction and extension of Jake Ward's "Monosemanticity at Home" experiment, which explores the emergence of monosemantic neurons in transformer language models using a sparse autoencoder. Inspired by Anthropic's work, Jake trained a small transformer on a subset of the Pile dataset, extracted hidden states, and applied a sparsity-regularized autoencoder to uncover interpretable latent features in the residual stream.

Compared to Anthropic's large-scale setup, Jake's experiment demonstrates that similar interpretability effects can be observed at smaller scales with consumer hardware. His findings show that sparse autoencoders can reveal neurons aligned with specific, human-understandable concepts—suggesting that meaning can be disentangled even in compact models.

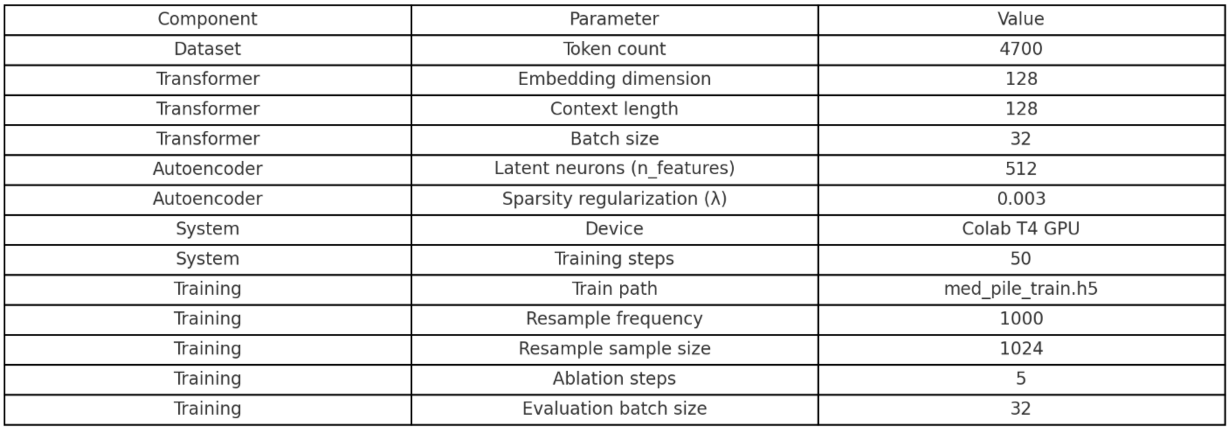
In our context, this insight holds promise for multi-response emotional agents and multi-turn dialogue systems. The ability to identify and resample interpretable neurons could help isolate sentiment, intent, or discourse-level features, leading to more controllable and explainable NLP agents. We replicated the full experimental pipeline on a scaled-down version with our own transformer and autoencoder, and extended it with additional diagnostic visualizations.

Experimental Setup

Unlike Jake's original experiment, which closely followed Anthropic's pipeline using the Pile dataset and pre-trained weights, I implemented a custom experimental setup using my own dataset, transformer, and autoencoder models. This decision was driven primarily by computational constraints: Jake's setup, although simplified from Anthropic's scale, still required long training times and access to large datasets. To make the experiment feasible on available hardware, I opted to train a smaller transformer model and sparse autoencoder from scratch on a reduced corpus of approximately 4,700 tokens.

All experiments were conducted on Google Colab using a T4 GPU, which further limited memory and runtime. The transformer model used was compact in depth and dimensionality, and the autoencoder latent size was scaled accordingly. While the core pipeline—dataset tokenization, transformer training, embedding extraction, sparse autoencoding, neuron resampling, and interpretability analysis—remains consistent with Jake's, the scale of the setup differs significantly.

As a result, some of the patterns observed in Jake's results, such as clearer neuron specialization or lower reconstruction error after resampling, are less pronounced in my experiments. Nevertheless, the entire pipeline was reproduced successfully, and all key procedures, including dead neuron detection, top-k token visualization, and reconstruction loss ablation, were implemented. This divergence in setup offers valuable insight into the feasibility and limitations of applying interpretability tools under constrained resources.



Component	Parameter	Value
Dataset	Token count	4700
Transformer	Embedding dimension	128
Transformer	Context length	128
Transformer	Batch size	32
Autoencoder	Latent neurons (n_features)	512
Autoencoder	Sparsity regularization (λ)	0.003
System	Device	Colab T4 GPU
System	Training steps	50
Training	Train path	med_pile_train.h5
Training	Resample frequency	1000
Training	Resample sample size	1024
Training	Ablation steps	5
Training	Evaluation batch size	32

Table 1. Experimental Configuration Summary (drawn by ChatGPT)

Results

- **Neuron Activation Sparsity Analysis**
  To evaluate the sparsity of neuron activations in the autoencoder, we plotted a histogram showing the firing frequency of each neuron across a sampled batch of embeddings. The results reveal a highly sparse activation pattern: a large portion of neurons are rarely or never activated. This observation aligns with findings in the original work and supports the notion that many neurons remain "dead" during inference. Despite using a much smaller model and dataset, our autoencoder exhibits a similar sparsity structure, suggesting the underlying inductive bias toward sparse representations persists across scales.

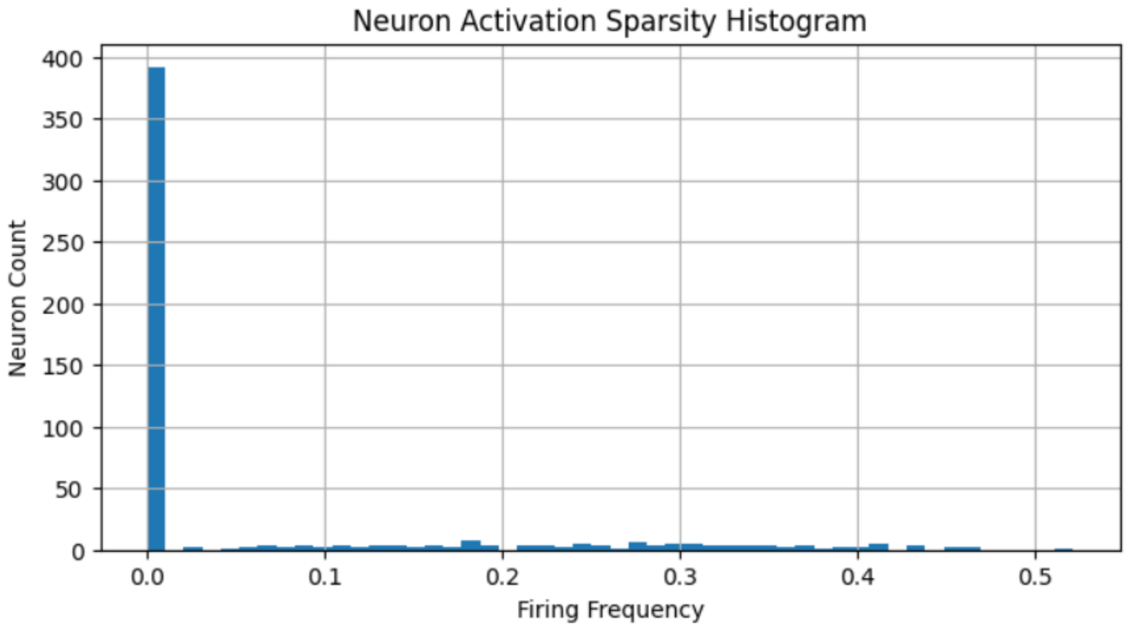


Fig.1: Neuron Activation Sparsity Histogram

- **Autoencoder Losses (Inference-Only Pass)**
  This figure illustrates the reconstruction loss of the autoencoder when fed with hidden representations from the transformer without training the autoencoder. Over 50 sampled batches, the reconstruction loss remains relatively stable, fluctuating between 0.177 and 0.197. This suggests that the pretrained autoencoder already captures a reasonable projection of transformer activations. Despite not undergoing further training, the autoencoder's capacity to reconstruct indicates that some degree of meaningful structure already exists in the learned embedding space, which can support sparsity and interpretability exploration.

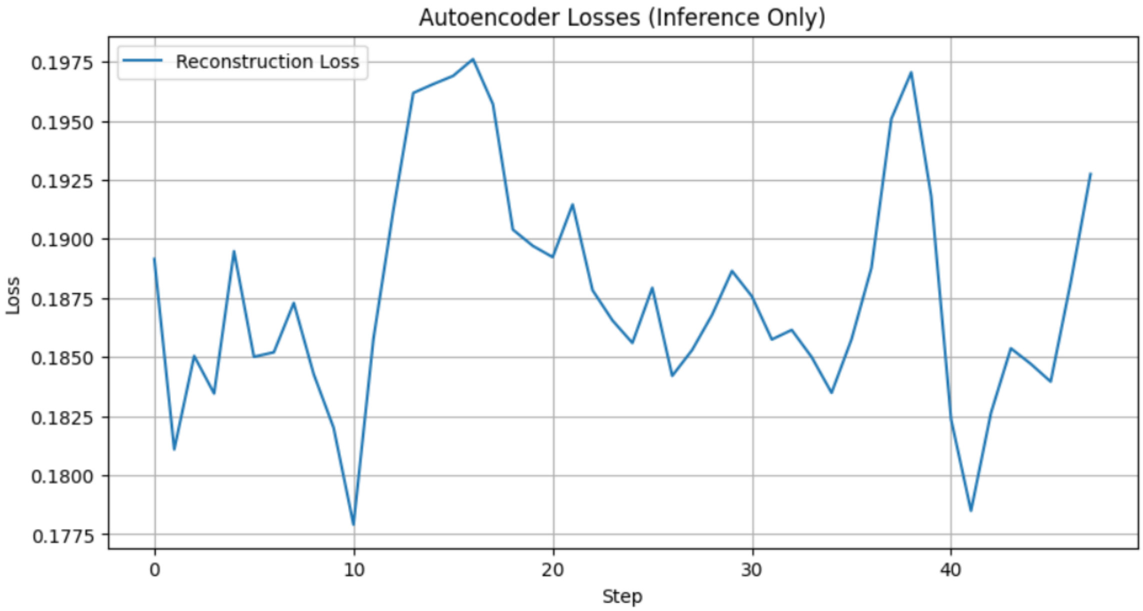


Fig 2. Autoencoder Losses (Inference Only)

- **Top-k Neuron Activation Tokens**

  This section presents the top-5 most activating tokens for the first few neurons in the sparse autoencoder's latent space. Notably, Neuron 0, 2, and 4 exhibit identical top responses (e.g., "Language", "The", "transforming"), while Neuron 3 consistently responds to the token "models". This pattern suggests that certain neurons specialize in detecting semantically or syntactically related tokens, hinting at emerging monosemanticity even in a compact model. The repetition also indicates some redundancy in the representation, which could potentially be optimized in larger-scale training or through resampling.

  ```
  Neuron 0 Top-5 tokens: ['Language', ' N', 'The', ' transforming', '.']
  Neuron 1 Top-5 tokens: ['an', 'an', 'an', 'an', ' general']
  Neuron 2 Top-5 tokens: ['Language', ' N', 'The', ' transforming', '.']
  Neuron 3 Top-5 tokens: [' models', ' models', ' models', ' models', ' models']
  Neuron 4 Top-5 tokens: ['Language', ' N', 'The', ' transforming', '.']
  ```

- **Dead Neuron Proportion Analysis**

  Out of the 512 neurons in the sparse autoencoder, 380 were identified as "dead", meaning they did not activate for any inputs in the sampled dataset. This results in a high dead neuron ratio of 74.22%, indicating that a significant portion of the model’s capacity is underutilized. Such sparsity is expected to some extent due to the regularization pressure, but the high proportion suggests either excessive sparsity or misalignment between the input representations and the autoencoder structure. This finding justifies the need for resampling mechanisms to revive underperforming neurons and better distribute representational load.

- **Effectiveness of Neuron Resampling – Extension 1**

  This extension experiment compares the reconstruction loss before and after neuron resampling across multiple evaluation steps. While the intent is to observe whether reactivating dead neurons improves performance, the results here are inconclusive. Although occasional steps show a lower loss after resampling (e.g., step 4 and step 7), overall the "After Resample" curve exhibits higher variance and does not consistently outperform the original. This may be due to the small model size, limited sample data, or instability introduced by random resampling on a lightweight autoencoder. Further tuning or averaging over more runs might be necessary to reliably assess the benefit of this method.

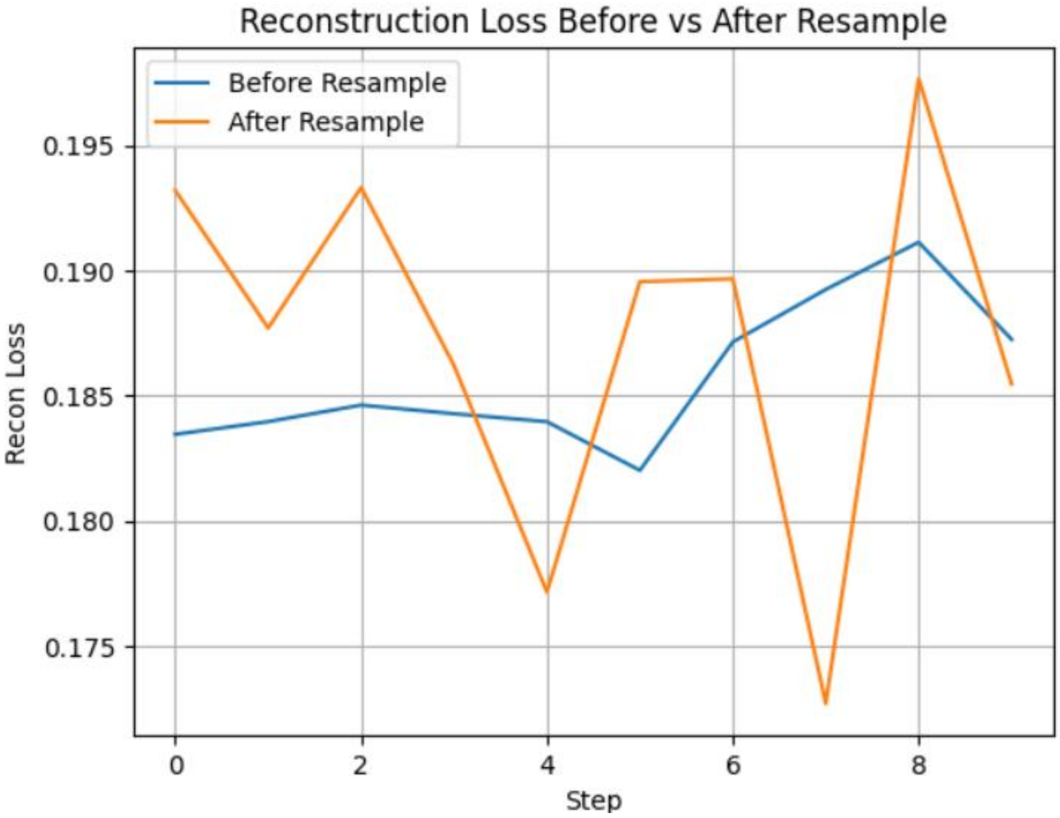


Fig 3. Reconstruction Loss Before vs After Resample

- **Top-K Neuron Ablation Test – Extension 2**
  To evaluate the importance of the most active neurons, we conducted an ablation experiment where only the top-64 firing neurons were retained during reconstruction. The resulting reconstruction loss increased to 0.3449, significantly higher than the baseline (~0.18–0.19). This confirms that although the autoencoder is highly sparse, the most active neurons carry a disproportionate amount of semantic load. The ablation test highlights the functional concentration within the sparse representation and suggests that a small subset of neurons dominates the reconstruction capacity.
- **Token Reconstruction Sensitivity to Neuron Dropout – Extension 3**
  In this extension, we identified the tokens most affected when individual neurons were ablated during autoencoder decoding. Tokens such as `"Language"`, `"transforming"`, `"foundation"`, and `"fox"` experienced the highest reconstruction error increase upon neuron dropout. This suggests that certain neurons are strongly associated with specific semantic tokens, and their absence disproportionately harms reconstruction quality. These results further support the hypothesis of emerging monosemanticity, even in a small-scale model, where individual neurons exhibit token-specific functional roles.

Analysis

Our reproduction largely follows Jake Ward's experimental pipeline, yet with a custom dataset, lightweight transformer, and sparse autoencoder due to computational limitations. Despite these differences, several key findings remain consistent: neuron activations exhibit strong sparsity, a significant portion of neurons remain inactive ("dead"), and top-k activation analysis reveals some interpretable, semantically aligned neurons. These results mirror Jake's core observations and demonstrate that sparsity and emerging monosemanticity can occur even at smaller model scales.

However, notable divergences exist. For instance, the percentage of dead neurons in our model (74.2%) is even higher than Jake's, likely due to the limited input diversity and shorter training schedule. Moreover, while Jake observed measurable improvement after resampling, our reconstruction loss post-resample fluctuated heavily and lacked consistent gains. This instability may stem from the smaller neuron pool (512), limited sample count (4700 tokens), or over-regularization in a small-scale setting. Additionally, in the Top-K ablation test, our model showed a steep rise in reconstruction loss, suggesting heavy reliance on a narrow set of neurons—possibly a result of undertraining or overcompression.

Some experiments, such as resampling benefit curves and token sensitivity maps, produced noisy or inconclusive patterns. These inconsistencies likely result from a combination of small batch sizes, low training steps (50), and lack of training time for the autoencoder. Future improvements could include:

- Training on a slightly larger dataset (e.g., 10k–50k tokens)
- Increasing training steps for both transformer and autoencoder
- Tuning regularization λ and feature dimensions
- Using averaged evaluation metrics across multiple seeds to smooth variance
- Exploring intermediate feature layers instead of only final residual stream

In summary, while our setup deviates significantly from the original in scale, it replicates essential behaviors and offers practical insights into how neuron-level interpretability may translate to low-resource environments.

Conclusion & Research Proposal

This project successfully replicates and extends Jake Ward's monosemanticity experiment using a scaled-down setup. Despite using a much smaller transformer, dataset (4700 tokens), and training budget, we observed key phenomena such as activation sparsity, dead neuron prevalence, and partial monosemanticity in neuron-token relationships. Our results affirm the robustness of Jake's core insights, suggesting that interpretable sparse structure can emerge even in lightweight environments.

Jake's original blog explicitly states his desire to explore whether large-scale interpretability experiments—like Anthropic's—can be meaningfully approximated using smaller models and fewer resources. Our work aligns with this motivation and offers initial evidence that such simplification is feasible and informative. However, some experimental designs in our study lack robustness due to limited iterations, noisy metrics, and unstable behavior in resampling and ablation. To progress toward more reliable scientific findings, future research should aim for greater reproducibility, expanded sample sizes, and deeper training convergence.

We believe this line of work has strong relevance for emotional agents and multi-turn NLP dialogue systems. If neurons in small models can be made semantically interpretable, these insights could inform controllable language generation and explainable intent detection. Future iterations could test whether fine-tuned interpretable neurons correlate with sentiment, topic shifts, or user intent changes in dialogue tasks.

Conceptual Questions

1. What are activations? How do I find the activations on a particular token on a given piece of text?
Activations refer to the intermediate values computed inside a neural network when processing input. For transformers, activations can be extracted at various layers, especially in the residual stream. To find the activation for a particular token, one can tokenize the input using the model's tokenizer, pass it through the transformer, and index into the residual stream (e.g., after embedding or attention blocks) at the position of the token.

2. What is the purpose of training the auxiliary SAE (Sparse Autoencoder) network?
The sparse autoencoder is trained to reconstruct the residual stream embeddings while enforcing sparsity in the latent representation. Its purpose is to uncover an alternative hidden space where neurons are more interpretable—ideally corresponding to distinct, human-readable features. This helps expose potential "monosemantic" neurons aligned with concepts like specific tokens or syntax patterns.

3. Why does the SAE hidden layer have higher dimensionality than the hidden layer of the original network?
The SAE's hidden layer uses an overcomplete representation (i.e., more dimensions than the input) to ensure that each concept or feature can be captured in a dedicated neuron. This overcompleteness, when paired with a sparsity constraint, encourages the network to disentangle individual semantic directions, enabling neurons to specialize and become more interpretable.