Evaluating Synthetic Activations composed of SAE Latents in GPT-2

Giorgi Giglemiani *
LASR Labs
giglemiani@gmail.com

Nora Petrova *
LASR Labs
nora.axion@gmail.com

Chatrik Singh Mangat *
LASR Labs
chatrikmangat@outlook.com

Jett Janiak
LASR Labs
jettjaniak@gmail.com

Stefan Heimersheim
Apollo Research
stefan@apolloresearch.ai

Abstract

Sparse Auto-Encoders (SAEs) are commonly employed in mechanistic interpretability to decompose the residual stream into monosemantic SAE latents. Recent work demonstrates that perturbing a model's activations at an early layer results in a step-function-like change in the model's final layer activations. Furthermore, the model's sensitivity to this perturbation differs between model-generated (real) activations and random activations. In our study, we assess model sensitivity in order to compare real activations to synthetic activations composed of SAE latents. Our findings indicate that synthetic activations closely resemble real activations when we control for the sparsity and cosine similarity of the constituent SAE latents. This suggests that real activations cannot be explained by a simple "bag of SAE latents" lacking internal structure, and instead suggests that SAE latents possess significant geometric and statistical properties. Notably, we observe that our synthetic activations exhibit less pronounced activation plateaus compared to those typically surrounding real activations.

1 Introduction

Neural networks often exhibit polysemanticity, where individual neurons fire for multiple features [Olah et al., 2017]. This complexity poses a significant challenge for interpreting computation in models, as it obscures the direct relationship between neuronal activations and specific semantic concepts. To explain polysemanticity, the theory of *superposition* suggests that neural networks represent more features than they have dimensions. These features are linearly represented as directions in activation space which form an overcomplete basis [Elhage et al., 2022, Bricken et al., 2023]. Although there is evidence for many linearly represented features, the claim that all features of a neural network are represented directions remains more speculative [Engels et al., 2024, Smith, 2024, Olah, 2024].

Sparse Auto-Encoders (SAEs) decompose a model's residual stream into a sparse set of features [Sharkey et al., 2022]. SAEs have become increasingly popular as a method to represent model activations in terms of more monosemantic and interpretable latents [Bricken et al., 2023, Cunningham et al., 2023, Templeton et al., 2024]. As the community increases its reliance on SAEs to interpret model behavior, it becomes more important to verify that decompositions generated by SAEs accurately capture abstractions that are used by the model.

*Equal contribution. Correspondence to giglemiani@gmail.com

Workshop on Attributing Model Behavior at Scale at NeurIPS 2024.

Neural networks employing superposition to represent features must address the challenge of interference to maintain performance [Hänni et al., 2024]. This necessitates an ability to accurately extract individual features while mitigating noise from "nearby" features in the representation space. Heimersheim and Mendel [2024] observed two key phenomena related to this: activation plateaus and directional sensitivity. Both are characterized by changes in the L2 distance of model activations at the final layer in response to early layer perturbations. Activation plateaus represent regions around an activation within which perturbations do not affect the model output significantly, indicating model robustness to small amounts of noise. Importantly, these activation plateaus are present around model-generated activations (real activations) but not around random points sampled from the distribution of model-generated activations (random activations). This suggests the presence of an error correction mechanism that the model might be using to deal with interference. Directional sensitivity refers to the variation in the model's response to perturbations based on the direction of the noise being added [Lindsey, 2024, Heimersheim and Mendel, 2024]. The model's sensitivity to a particular direction is characterized by the inverse of the perturbation magnitude required in that direction to induce a step-function-like change (blowup) in the model output.

In this paper, we focus on generating synthetic activations composed of SAE latents and testing whether they behave like real activations. We specifically test the "bag of SAE latents" hypothesis and discover that arbitrary combinations of SAE latents do not produce activations that resemble real activations. We find that controlling for the sparsity and the cosine similarity of SAE latents enables us to create synthetic activations that most closely resemble the sensitivity of real activations. We test whether these observations generalize to activation plateaus and find that synthetic activations don't behave like real activations.

Our contributions are:

1. We find that the "bag of SAE latents" approach is not sufficient to produce synthetic activations that resemble model-generated (real) activations.
2. We find that the sparsity of the top SAE latent, the relative latent activations, and the cosine similarity between the active latents and the top latent play an important role in determining whether synthetic activations behave like real activations.
3. The performance of synthetic activations in the sensitivity experiment does not transfer to the activation plateau experiment that we conduct. We find that synthetic activations do not have activation plateaus around them like real activations do.

2 Background

Our experiments are based on the setup described in Heimersheim and Mendel [2024], wherein they perturbed model activations at an early layer and measured the effect it had on the L2 distance of late-layer activations. They investigated activation plateaus and sensitive directions in `GPT-2`, motivated by the error correction mechanism predicted by computation in superposition. They explored two key predictions: (1) model-generated activations should be resistant to small perturbations, exhibiting "activation plateaus", and (2) perturbations towards model-generated activations should affect model output more quickly than towards random directions. Their findings supported both of their predictions, providing evidence for an error correction mechanism used by the model to suppress small amounts of noise. This research aimed to better understand computation in superposition and to find dataset-independent evidence for model features, potentially connecting to SAE research.

3 Related Work

Previous works have studied the model's response to residual stream perturbations using different experiments. We discuss the works that are the most relevant to ours below.

Janiak et al. [2024] identified stable regions (corresponding to activation plateaus) in the activation space of transformer-based models, hypothesizing their role in error correction and semantic distinctions. Our work primarily focuses on sensitive directions, though we study activation plateaus around synthetic activations and compare them against real activations.

Gurnee [2024] found that substituting the model activation at an early layer with its SAE reconstruction causes a bigger jump in KL divergence of the model's next-token prediction probabilities than

substituting it with a random vector that is the same L2 distance away from the original activation as the SAE reconstruction. While our work focuses on compositions of SAE latents, we study the effect of SAE reconstruction error on our experiments (Appendix D).

Lee and Heimersheim [2024] found that SAE reconstruction errors are not pathologically large when compared to more realistic baselines. They also found that end-to-end SAE latents do not exhibit stronger effects on model output compared to traditional SAE latents. Their work focuses on perturbations in individual SAE latent directions while we study compositions of SAE latents.

Lindsey [2024] found that ablating an SAE latent had a significantly larger effect on model performance than doubling the latent's activation. Additionally, they found that dampening latent activations had almost as strong of an effect on the output distribution as latent ablation. In our study, we focus on composing synthetic activations and studying SAE latent properties.

4 Method

Since model-generated (real) activations exhibit distinct behaviors in both experimental settings used in Heimersheim and Mendel [2024], we test whether synthetic activations composed of SAE latents do the same. We adapt these experimental settings to study relationships between SAE latents that are important for generating synthetic activations that are in-distribution for the model. We discuss the methodology for the directional sensitivity experiment in Section 4.1, and discuss the different types of activations we test in Sections 4.2 and 4.3. Finally, we discuss the activation plateau experiment in Section 4.4.

4.1 Perturbation Setup

For each perturbation, we only make changes to the activations at the last token position in the context at layer 1 (`blocks.1.hook_resid_pre`). We refer to the original unperturbed activation as the base activation, often denoted by A. We perturb the base activation A by slowly adding increments towards a direction D:

$$\mathrm{A}_{\mathrm{pert}}(\mathrm{n}) = \mathrm{A} + 0.5 \cdot \mathrm{n} \cdot \mathrm{D}$$

where n is the step number, going from 0 to 100, and D is a unit vector. For all our perturbations, we define the direction D as the normalised difference between a base activation and a target activation. We use a step size of 0.5 because this makes the perturbation norm comparable to the typical norm of activations, which is $\simeq 56$.

For each perturbation, we compute the L2 distance between the activations of the original and the perturbed run after the final layer (`blocks.11.hook_resid_post`). We use this L2 distance to study the effect of our perturbations instead of the KL divergence of the next-token prediction probabilities. We focus on L2 distance because KL divergence plots tend to obscure the structure of activation plateaus (we still include KL divergence based results in Appendix C for comparison to previous work).

We test multiple metrics in order to formalize the location of the blowup in L2 distance, and we find that the maximum slope of the L2 distance against the perturbation step curve represents it most cleanly (shown in Figure 1; see Appendix B for the other metrics we test). For each perturbation, we label the maximum slope (MS) step and use it to refer to the location of the blowup further in this work.

For all our experiments, we use `GPT2-small` [Radford et al., 2019] and run inference on randomly sampled prompts of sequence length 10 from the `OpenWebText` dataset* [Gokaslan and Cohen, 2019], and we collect model-generated activations from the residual stream at Layer 1 (`blocks.1.hook_resid_pre`). We use `GPT2-small` SAEs [Bloom, 2024], `sae-lens` [Bloom and Chanin, 2024] and `TransformerLens` [Nanda and Bloom, 2022] to perform our experiments and generate synthetic activations.

*Using a tokenized version of this dataset available here.

4.2 Non-SAE Baselines

In order to compare our setup to previous work [Heimersheim and Mendel, 2024], we run perturbations towards model-generated (real) and random activations. We sample 1000 prompts from the dataset and run inference on them to obtain the base activations, and perturb each base activation in two directions:

1. **Model-generated (real):** Towards a randomly selected activation produced by the model.
2. **Random:** Towards a randomly sampled point from a normal distribution with the same mean and covariance as model-generated activations (calculated using $32,000$ model-generated activations).

We plot examples of perturbations towards real and random activations in Figure 1. The distribution of distances between the base activation and the target are similar for both baselines, with a mean of $\simeq 40$ in activation space. Additionally, we find that the average cosine similarity between two model-generated activations with respect to the SAE decoder bias is $\simeq 0.42$.

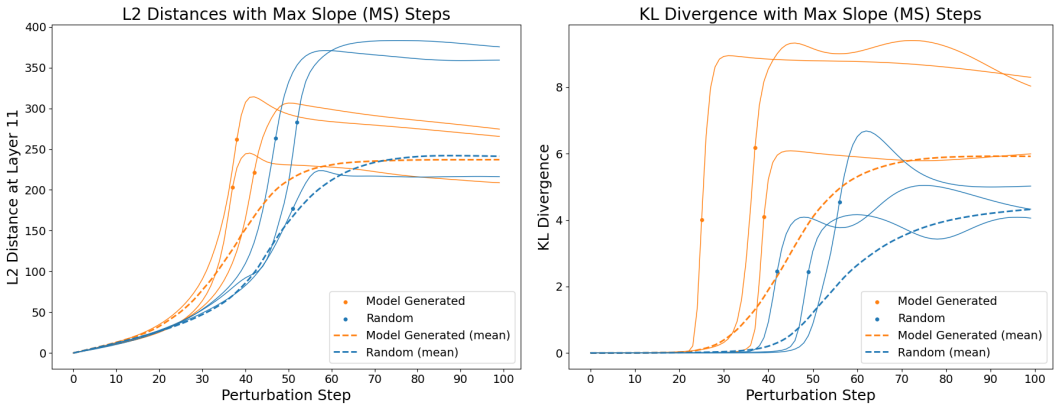

Figure 1: The L2 distance after Layer 11 (left) and the KL divergence of the next-token prediction probabilities (right) between the perturbed and unperturbed model, as three base activations at Layer 1 are slowly perturbed towards model-generated activations (orange) and random points sampled from the distribution of model activations (blue). The x-axis represents the total length of the perturbation broken into 100 steps of size 0.5 each. The dot on each solid line represents the maximum slope (MS) step for each perturbation. The dashed lines represent the average L2 distance and KL divergence per step for 1000 perturbations of both types. The linear part at the start of the curves represents the activation plateau, and the sharp rise in the curves represents the blowup.

4.3 Synthetic Activations

We construct synthetic activations using three methods that each use different amounts of information about SAE latents, and compare which ones match the behavior of real activations.

Our simplest approach (**synthetic-random**) randomly selects SAE latents and assigns them the same latent activations as the active latents of the base activation. We consider this a weak approach to compose synthetic activations and mainly focus on the next two methods, but we show the results for this method in Appendix A.

Our second approach (**synthetic-baseline**) accounts for SAE latent sparsities along with latent activations, matching the distribution of latent sparsities from the base activation:

1. We use the SAE to encode the base activation and obtain its active latents and their latent activations.
2. We replace each active SAE latent with a new one sampled from 10 most similarly-sparse latents and assign it the latent activation of the original active latent.
3. We decode the new latent activations to obtain the **synthetic-baseline** activation.

For our third and final method (**synthetic-structured**) we capture and reproduce geometric properties that SAE latents of real activations have. We get the best results when we control for sparsity of latents and the cosine similarity relationship between latents. The procedure for the generation of synthetic-structured activations is as follows:

1. We use the SAE to encode the base activation and obtain its active latents and their latent activations. The active latent with the highest latent activation is the top latent of the base activation (`top_base`).
2. We create a list of 100 non-dead SAE latents with the most similar sparsity to `top_base`.
3. Out of the 100 selected latents, we select one latent that has cosine similarity closest to 0.42 (mean cosine similarity between two real activations w.r.t. the SAE decoder bias) with `top_base`.
4. This latent becomes the top latent for our synthetic activation (`top_synth`), and we give it a latent activation value equal to that of `top_base`.
5. For each remaining active latent (`l_base`) in the base activation:
   (a) We calculate its cosine similarity (`l_top_cos_sim`) with `top_base`.
   (b) We select a latent (`l_synth`) that has cosine similarity with `top_synth` equal to `l_top_cos_sim`.
   (c) We assign `l_synth` a latent activation value equal to that of `l_base`.
6. We construct a latent activation vector with zeros for all latents except the latents selected above, and decode it to obtain the **synthetic-structured** activation.

We perform 1000 perturbations towards synthetic-baseline and synthetic-structured activations each, using the setup described in Section 4.1, where the direction D is the normalised difference between the synthetic activation and the base activation.

4.4 Activation Plateaus

If our synthetic activations behave like real activations, they should not only reproduce directional sensitivity behavior, but also exhibit activation plateaus. To test this, we use the following perturbation approach:

1. We initiate perturbations from four distinct base activations: model-generated activations, synthetic activations generated using SAE latents of similar sparsity to the base activation (synthetic-baseline), synthetic activations generated using selected SAE latents based on sparsity and cosine similarity relationship (synthetic-structured), and random points sampled from the distribution described in Section 4.2 (random).
2. For all four starting points, we perturb towards a random activation.
3. We track how quickly L2 distance at Layer 11 increases near the start of the perturbation by recording the activation plateau perturbation step (AP step) at which the L2 distance between the unperturbed and perturbed models crosses a value of 20 (we found that a simple threshold was enough to distinguish the behavior of different activations).

The AP step measures the flatness of the activation plateau around activations, with larger AP steps signify flatter activation plateaus. We repeat this process for 1000 perturbations and collect the distributions of AP steps for each activation type.

5 Results

We find that the behavior of synthetic activations as compared to random activations and real activations differs for the two experiments we perform. This suggests that directional sensitivity and activation plateaus point to different properties of SAE latents that make up real activations (see Appendix E for details on properties that we account for). We now provide detailed results for each of our experiments below.

5.1 Directional Sensitivity

For this experiment, we perturb real activations towards different types of activations we construct and study the model's sensitivity to these perturbations. Figure 2 shows the distributions of max slope (MS) steps for L2 distance for perturbations towards synthetic-baseline and synthetic-structured activations compared to model-generated (real) and random activations. We also provide statistics for the MS step distributions for perturbations towards all activations types in Table 1.

When comparing the MS steps (Table 1) for perturbations towards real and random activations, we find that perturbations towards real activations cause earlier (lower mean) and more localized (lower variance) blowups than perturbations towards random activations. This means that perturbing towards real activations affects the model output more than perturbing towards random activations.

While synthetic-baseline activations do not fully replicate the behavior of real activations, they still perform better than random activations (Figure 2). We calculate similarity between two MS step distributions using the Kolmogorov Smirnov (KS) statistic [Smirnov, 1948]. This demonstrates that the model is more sensitive to perturbations towards synthetic activations composed of SAE latents than perturbations towards randomly sampled points from the distribution of model activations (random). This suggests that SAE latents encode more information about model computation than random directions do.

Importantly, perturbations towards synthetic-structured activations look a lot more similar to perturbations towards model-generated activations than perturbations towards synthetic-baseline and random activations do (Figure 2, Table 1). This implies that relationships between SAE latents are important, and that model-generated activations are not approximated well by "bags of SAE latents". Additionally, we find that synthetic-random activations perform worse than synthetic-baseline activations, confirming our decision to use the latter as a stronger baseline (see Appendix A for more details on synthetic-random activations).

The distance between the base activation and the target activation varies for each perturbation we perform, and we find that this can directly influence the location of the blowup in our setup. In order to remove the effect of the distance on the MS step distribution, we also perform perturbations with relative step size (see Appendix A for details). The gap between the synthetic-structured and synthetic-baseline reduces in the relative step size setup. This is because synthetic-baseline activations are further away from base activations than synthetic-structured activations, and hence cause blowups later.

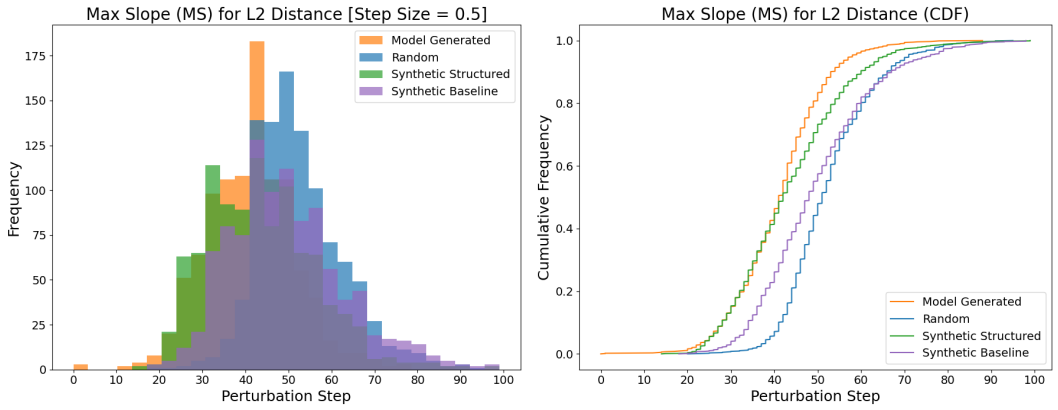

Figure 2: The distributions of the max slope (MS) steps for perturbations towards model-generated (orange), random (blue), synthetic-baseline (purple), and synthetic-structured (green) activations. The left panel shows the counts of MS steps occurring in different bins along the length of the perturbation, and the right panel shows corresponding cumulative frequency. We find that perturbing towards synthetic-structured activations is more similar to perturbing towards model-generated activations as compared to perturbing towards synthetic-baseline activations.

Max Slope (MS) step distribution statistics

Activation Type	Mean	Std dev	KS
Model Generated	41.11	10.40	0.00
Random	52.49	**10.21**	0.45
Synthetic Baseline	49.61	13.25	0.28
Synthetic Structured	**43.48**	12.79	**0.11**

Table 1: When comparing perturbations towards different activations, we find that synthetic-structured activations behave more similar to model-generated activations than synthetic-baseline and random activations do. This table contains the mean, standard deviation and KS statistic for MS step distributions for all the perturbations we perform with fixed step size. The KS statistic is measured against perturbations towards model-generated activations, with lower values indicating higher similarity.

5.2 Activation Plateaus

For this experiment, we perturb different types of activations towards random directions to assess whether they have activation plateaus around them. Figure 3 shows the distributions of AP steps for L2 distance for perturbations starting at synthetic-baseline and synthetic-structured activations compared to model-generated and random activations.

Our findings reveal varying sizes of plateaus around different types of activations. Model-generated activations exhibit more pronounced plateaus, indicating greater robustness to noise compared to synthetic and random activations, which display less distinct plateau regions. The relatively less pronounced plateaus around synthetic-baseline activations provide additional evidence against the "bag of SAE latents" approach, as this behavior notably differs from that of model-generated activations. While synthetic-structured activations show improvement, the differences in plateau characteristics suggests that they may not fully capture all significant relationships between SAE latents. To quantify the impact of SAE reconstruction error on the discrepancy between synthetic and model-generated activations, we conducted tests detailed in Appendix D, which indicate that this contribution is minimal.

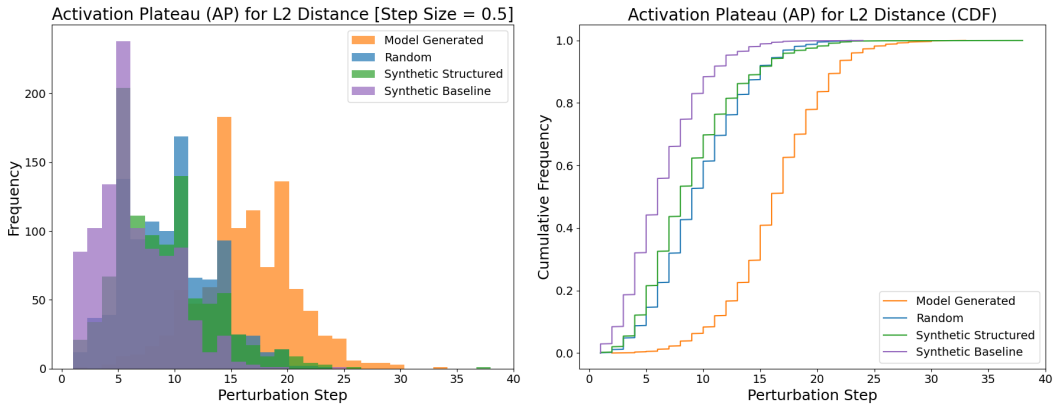

Figure 3: The distributions of the activation plateau (AP) steps for perturbations starting at model-generated, random, synthetic-baseline, and synthetic-structured activations. We perturb towards random activations in all cases. The left panel shows the counts of AP steps occurring in different bins along the length of the perturbation, and the right panel shows the cumulative frequency for the same. We find that model-generated activations (orange) have flatter plateaus around them than all of the other activation types. We also see that synthetic-baseline activations (purple) have the steepest plateaus around them, while plateaus around synthetic-structured (green) and random (blue) activations look similar.

6 Limitations

The heuristics we use to construct synthetic activations leave room for improvement, as evidenced by the gap between them and model-generated activations, especially for activation plateaus. We use

cosine similarity between SAE latents to capture geometric relationships between them, but leave accounting for latent co-occurrence and other relationships between latents for future work.

We find that the synthetic activations we construct do not match the cosine similarity distribution of SAE latents in model-generated activations closely (see Appendix E). Our method also leverages information from the base activation in order to construct synthetic activations. While this is not ideal, we have verified that using information from a different model-generated activation for the construction does not change our results.

The L2 distance curves across perturbation steps exhibit significant variability, potentially impacting the effectiveness of our MS metric in identifying key steps. This metric assumes curve smoothness, which may not hold in practice (Figure 1). More robust metrics could potentially produce clearer results (see Appendix B for our exploration of other metrics).

Our study focuses solely on one layer of `GPT2-small`, necessitating further investigation across different layers, models, and SAEs to establish broader applicability. Additionally, we confined our perturbations to the final token position only. Exploring different context lengths is crucial to assess the generalizability of our findings. These extensions would provide a more comprehensive understanding of the observed phenomena.

7 Conclusion

We find additional evidence that `GPT-2` is more sensitive to perturbations towards model-generated activations than random directions, and that model-generated activations cannot simply be explained by "bags of SAE latents". We find that leveraging statistical and geometric properties of SAE latents helps us create synthetic-structured activations which are more similar to model-generated activations. Yet, these synthetic-structured activations lack the characteristic plateaus surrounding model-generated activations, suggesting there may be additional SAE latent properties influencing model computation.

This presents exciting avenues for future work on model sensitivity to perturbations:

- Coming up with better approaches to constructing synthetic activations that perform better in both experimental settings.
- Checking for the existence of thresholds below which the model output does not respond to changes in latent activations.
- Case studies which look at model sensitivity to perturbations in directions created using interpretable SAE latents and take context into account.
- Zooming into latent ablation based perturbations further to study which latents contribute the most to blowups.

8 Acknowledgements

Erin Robertson, Charlie Griffin and the whole LASR Labs team for making this research project possible. Joseph Bloom for comments and SAEs used for this work; Daniel Lee for helpful discussions and for pointing us towards using NL as a metric; Jake Mendel for fundamental discussions about this research direction; Lawrence Chan, Nicholas Goldowsky-Dill, Bilal Chughtai, Daniel Tan, David Krueger, Joe Needham, David Chanin, Tomáš Dulka and Diogo Cruz for review and comments.

References

Joseph Bloom and David Chanin. Saelens. `https://github.com/jbloomAus/SAELens`, 2024.

Joseph Isaac Bloom. Open source sparse autoencoders for all residual stream layers of gpt2-small, Feb 2024. URL `https://www.alignmentforum.org/posts/f9EgfLSurAiqRJySD/open-source-sparse-autoencoders-for-all-residual-stream`.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec,

Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models, 2023. URL https://arxiv.org/abs/2309.08600.

Nelson Elhage, Tristan Hume, Catherine Olsson, Neel Nanda, Tom Henighan, Scott Johnston, Sheer ElShowk, Nicholas Joseph, Nova DasSarma, Ben Mann, Danny Hernandez, Amanda Askell, Kamal Ndousse, Andy Jones, Dawn Drain, Anna Chen, Yuntao Bai, Deep Ganguli, Liane Lovitt, Zac Hatfield-Dodds, Jackson Kernion, Tom Conerly, Shauna Kravec, Stanislav Fort, Saurav Kadavath, Josh Jacobson, Eli Tran-Johnson, Jared Kaplan, Jack Clark, Tom Brown, Sam McCandlish, Dario Amodei, and Christopher Olah. Softmax linear units. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/solu/index.html.

Joshua Engels, Isaac Liao, Eric J. Michaud, Wes Gurnee, and Max Tegmark. Not all language model features are linear, 2024. URL https://arxiv.org/abs/2405.14860.

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

Wes Gurnee. Sae reconstruction errors are (empirically) pathological, Mar 2024. URL https://www.alignmentforum.org/posts/rZPiuFxESMxCDHe4B/sae-reconstruction-errors-are-empirically-pathological.

Stefan Heimersheim and Jake Mendel. Activation plateaus and sensitive directions in gpt2, Jul 2024. URL https://www.alignmentforum.org/posts/LajDyGyiyX8DNNsuF/interim-research-report-activation-plateaus-and-sensitive-1.

Kaarel Hänni, Jake Mendel, Dmitry Vaintrob, and Lawrence Chan. Mathematical models of computation in superposition, 2024. URL https://arxiv.org/abs/2408.05451.

Jett Janiak, Jacek Karwowski, Chatrik Singh Mangat, Giorgi Giglemiani, Nora Petrova, and Stefan Heimersheim. Characterizing stable regions in the residual stream of llms, 2024. URL https://arxiv.org/abs/2409.17113.

Daniel J. Lee and Stefan Heimersheim. Investigating sensitive directions in gpt-2: An improved baseline and comparative analysis of saes, Sep 2024. URL https://www.lesswrong.com/posts/dS5dSgwaDQRoWdTuu/investigating-sensitive-directions-in-gpt-2-an-improved.

Jack Lindsey. How strongly do dictionary learning features influence model behavior? *Transformer Circuits Thread*, 2024. https://transformer-circuits.pub/2024/april-update/index.html.

Neel Nanda and Joseph Bloom. Transformerlens. https://github.com/TransformerLensOrg/TransformerLens, 2022.

Chris Olah. What is a linear representation? what is a multidimensional feature? *Transformer Circuits Thread*, 2024. https://transformer-circuits.pub/2024/july-update/index.html.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization, 2017.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Lee Sharkey, Dan Braun, and Beren Millidge. Taking features out of superposition with sparse autoencoders, Dec 2022. URL https://www.alignmentforum.org/posts/z6QQJbtpkEAX3Aojj/interim-research-report-taking-features-out-of-superposition.

N. Smirnov. Table for Estimating the Goodness of Fit of Empirical Distributions. *The Annals of Mathematical Statistics*, 19(2):279 – 281, 1948. doi: 10.1214/aoms/1177730256. URL https://doi.org/10.1214/aoms/1177730256.

Lewis Smith. The 'strong' feature hypothesis could be wrong, Aug 2024. URL https://www.alignmentforum.org/posts/tojtPCCRpKLSHBdpn/the-strong-feature-hypothesis-could-be-wrong.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

A Analyzing different perturbations setups and synthetic activations

In the main paper we use absolute step size for perturbations, however blowup locations have a dependence on the distance between the base and target activations, which can make the MS step distributions with absolute step size misleading. We know that the blowup location does not solely depend on the distance between the base and target activations, and in order to isolate this property, we create a distance agnostic setup using relative step sizes. In the relative step size approach, our perturbations always start at a base activation (A) and end at a target activation (T) using linear interpolation:

$$\mathrm{A}_{\mathrm{pert}}(\mathrm{n}) = \left(1 - \frac{\mathrm{n}}{100}\right) \cdot \mathrm{A} + \frac{\mathrm{n}}{100} \cdot \mathrm{T}$$

where n is the perturbation step, which goes from 0 to 100. This method ensures that we always transition from the base activation to the target activation in a fixed number of steps, regardless of the distance between them. By using relative step size, we remove the dependence of the blowup location on distance, and instead compare the effect of perturbations purely in terms of the percentage of base and target activations present at each step. For example, step 50 in this setup implies that the perturbed activation is made up of 50% base activation and 50% target activation.

In the relative step size setup, we find that the MS step distribution for perturbations towards model-generated activations peaks more strongly around step 50 than in the absolute step size setup. The blowups are also localized between step 30 and 70, implying that blowups usually happen in the middle of the perturbation (Figure A.1). We posit that until step 30, the model treats the interpolated activation as the base activation. This is due to 70% of the interpolated activation coming from the base activation, and the remaining 30% coming from the target activation being treated as noise. This effect reverses at step 70, where the model starts treating the interpolated activation as the target activation, and the 30% that comes from the base activation is considered noise.

Our analysis reveals that the MS step distribution for random activation perturbations exhibits marginally higher variance than the absolute step size setup, with a rightward shift relative to the distribution for model-generated activation perturbations (Table A.1). This suggests that stronger perturbations towards random activations are required to induce a blowup compared to model-generated activations. Furthermore, it indicates that the model is more resilient to random noise than to noise directed towards another model-generated activation, requiring a greater magnitude of the former to cause confusion in the model.

In this setup, comparing perturbations with synthetic-baseline and synthetic-structured activations reveals that while synthetic-structured activations still more closely mimic model-generated activations, the disparity between the two has notably decreased (Figure A.1, Table A.1). This suggests that synthetic-baseline activations less effectively align with the residual stream geometry of model-generated activations compared to synthetic-structured ones, explaining the latter's superior performance in the absolute step size setup. Our findings indicate that considering latent sparsity is important for synthetic activations to emulate model-generated activations in the relative step size setup. Consequently, both synthetic-structured and synthetic-baseline outperform synthetic activations created using the "bag of SAE latents" approach without accounting for sparsity (synthetic-random).

We find that when we construct synthetic-structured activations (Section 4.3), omitting the cosine similarity constraint on the top latent and instead selecting based on sparsity similarity to the base activation's top latent yields the best-performing synthetic activations in the relative step size setup.

However, these activations typically have greater distance from the base activation compared to synthetic-structured activations. Consequently, their performance in the absolute step size scenario is inferior to that of synthetic-structured activations.

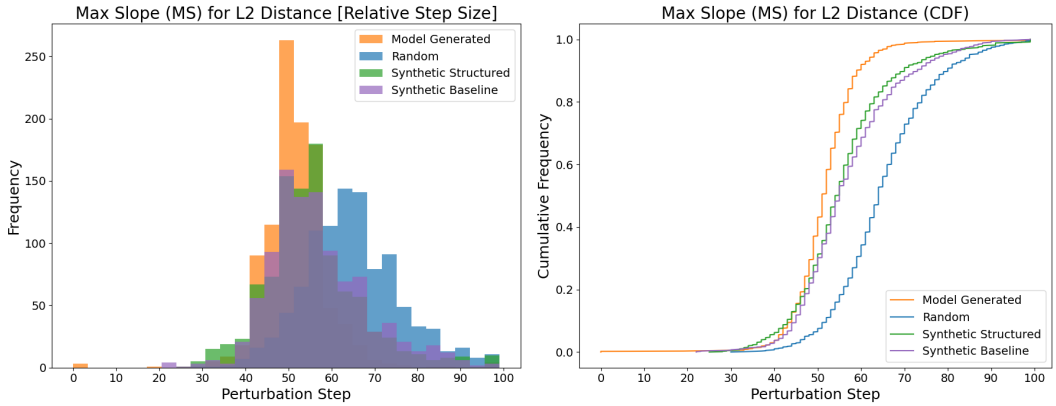

Figure A.1: The distributions of the max slope (MS) steps for perturbations with relative step size towards model-generated (orange), random (blue), synthetic-baseline (purple), and synthetic-structured (green) activations. The left panel shows the counts of MS steps occurring in different bins along the length of the perturbation, and the right panel shows the cumulative frequency for the same. We find that perturbing towards synthetic-structured activations in the relative step size setup is slightly more similar to perturbing towards model-generated activations than perturbing towards synthetic-baseline activations is.

Max Slope (MS) Step Distribution Statistics

Activation Type	Absolute Step Size			Relative Step Size		
	Mean	Std dev	KS	Mean	Std dev	KS
Model Generated	41.21	10.32	0.00	51.65	7.42	0.00
Random	52.49	10.34	0.44	64.89	10.50	0.62
Synthetic Baseline	49.88	12.60	0.31	57.07	11.29	0.27
Synthetic Structured	**43.45**	12.78	**0.11**	55.69	11.31	0.22
Synthetic Random	51.30	**10.25**	0.39	55.25	**8.74**	0.19
Synthetic Structured (w/o cos sim)	50.17	11.96	0.31	**54.47**	10.68	**0.17**

Table A.1: We find that controlling for the sparsity of the top latent and the cosine similarity between the active latents play an important role in making synthetic-structured activations perform well in both absolute and relative step setups. This table contains the mean, standard deviation and KS statistic for MS step distributions for all types of synthetic activations we tested. The KS statistic is measured against perturbations towards model-generated activations, with a lower value meaning higher similarity. The entries in bold show the best match with statistics for model-generated activations.

B Metrics for analysing blowups

In our main analysis, we focus on the maximum slope (MS) as an indicator of the blowup step. In this section we share findings using the Area Under the Curve (AUC) and Non Linear (NL) metrics to represent important parts of the L2 distance vs perturbation step curve.

B.1 Area Under Curve (AUC)

Our experimental results reveal that certain L2 distance curves deviate from the expected step-function-like pattern, causing the MS step to misrepresent the actual blowup location for these curves. In contrast, the AUC metric provides a more comprehensive assessment of activation behavior across the entire perturbation process. This approach not only identifies the steepest increase point but also effectively screens out atypical curves that might otherwise evade detection. AUC calculates the step at which the following ratio is maximized:

$$R = \text{area of the triangle defined by (0,0), (x,0) and (f(x),x)} / \text{area under the curve f(x)}$$

where $f(x)$ is L2 distance as a function of the perturbation step x. This method is sensitive to the concavity or convexity of the perturbation curve. For predominantly concave curves (where the rate of change increases over time), the AUC blowup step tends to occur later, as the triangular area takes longer to outpace the actual area under the curve. Conversely, for convex curves (where the rate of change decreases over time), the AUC blowup step tends to occur earlier. This property allows the AUC method to implicitly capture information about the shape of the perturbation.

The AUC metric serves as sanity check, confirming that most perturbations align with expectations. Convex L2 distance curves yield early AUC peaks, and Figure B.1 demonstrates that the majority of perturbations exhibit the anticipated concave shape. We find that our perturbation results hold for AUC step distributions in the absolute step size setup (Table B.1), with structured-synthetic activations more closely mimicking model-generated activations compared to synthetic-baseline activations. In the relative step size setup (detailed in Appendix A), synthetic-structured and synthetic-baseline activations perform similarly. This can be attributed to the higher prevalence of convex curves in perturbations towards synthetic-structured activations versus synthetic-baseline activations.

Area Under Curve (AUC) Step Distribution Statistics

Activation Type	Absolute Step Size			Relative Step Size		
	Mean	Std dev	KS	Mean	Std dev	KS
Model Generated	41.94	11.78	0.00	51.98	9.64	0.00
Random	52.73	13.66	0.43	64.97	16.01	0.59
Synthetic Baseline	49.31	14.20	0.25	56.66	13.20	0.22
Synthetic Structured	43.54	14.99	0.09	54.84	15.51	0.21

Table B.1: We find that our results for the AUC step distributions are similar to those for the MS step distributions. This table contains the mean, standard deviation and KS statistic for AUC step distributions for all the perturbations we perform. The KS statistic is measured against perturbations towards model-generated activations, with a lower value meaning higher similarity.

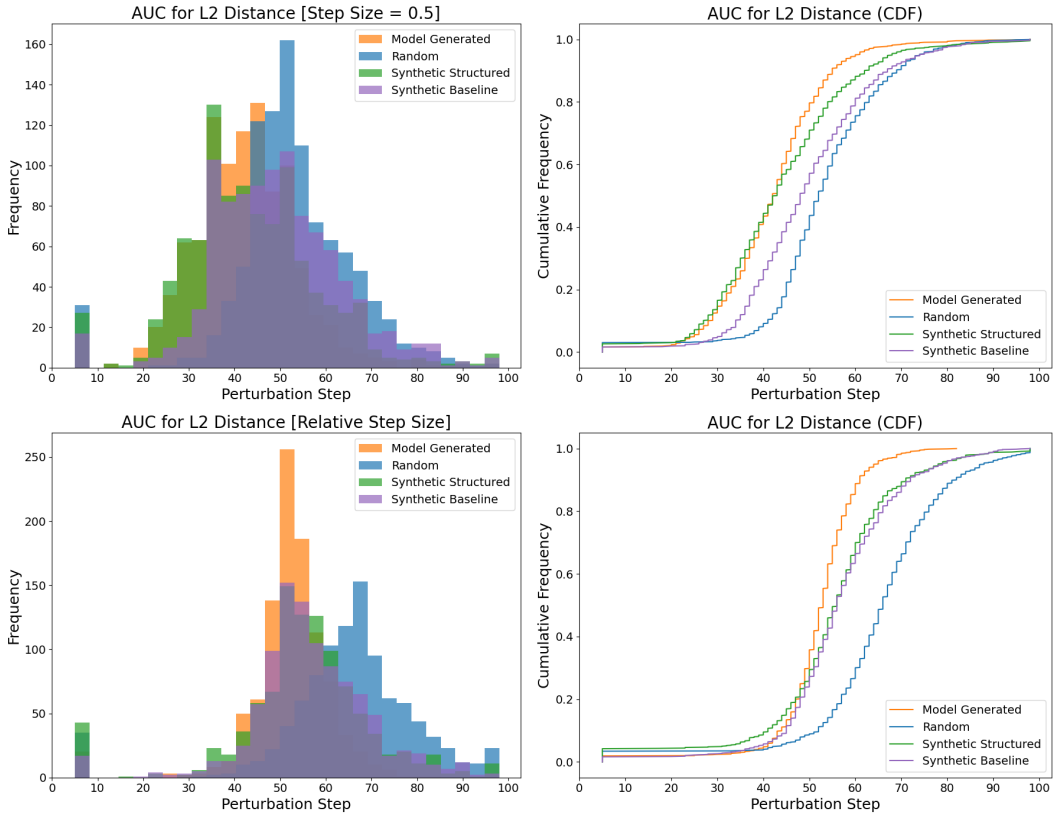

Figure B.1: The distributions of the AUC steps for perturbations with absolute step size (top) and relative step size (bottom) towards model-generated (orange), random (blue), synthetic-baseline (purple), and synthetic-structured (green) activations. The left column shows the counts of AUC steps occurring in different bins along the length of the perturbation, and the right column shows the cumulative frequency for the same. We find that our results for the AUC step distributions are similar to those for the MS step distributions.

B.2 Non-Linear (NL)

Using L2 distance to observe the perturbations reveals that the region before the blowup is not flat, but linear with varying slopes (Figure 1). In order to study the size of the initial linear portion of the curves, we use the Non-Linear (NL) metric, which points to the earliest step at which the slope of the L2 distance vs perturbation step curve deviates from linearity by more than 10% of the initial slope. We use this metric as an alternate measure for the size of the activation plateau around the base activation along different perturbation directions.

We observe that perturbations towards model-generated activations cause the quickest deviation from linearity followed by synthetic-structured activations, which is in line with our previous results for blowup locations (Figure B.2, Table B.2). However, we find that the deviation from linearity occurs the latest during perturbations towards synthetic-baseline activations, which suggests that L2 distance has a higher initial slope for these perturbations, giving more room for changes in the slope before they are classified as a deviation from linearity. In this case, the behavior of synthetic-baseline activations provides further evidence that local relationships between SAE latents are important to approximate model-generated activations.

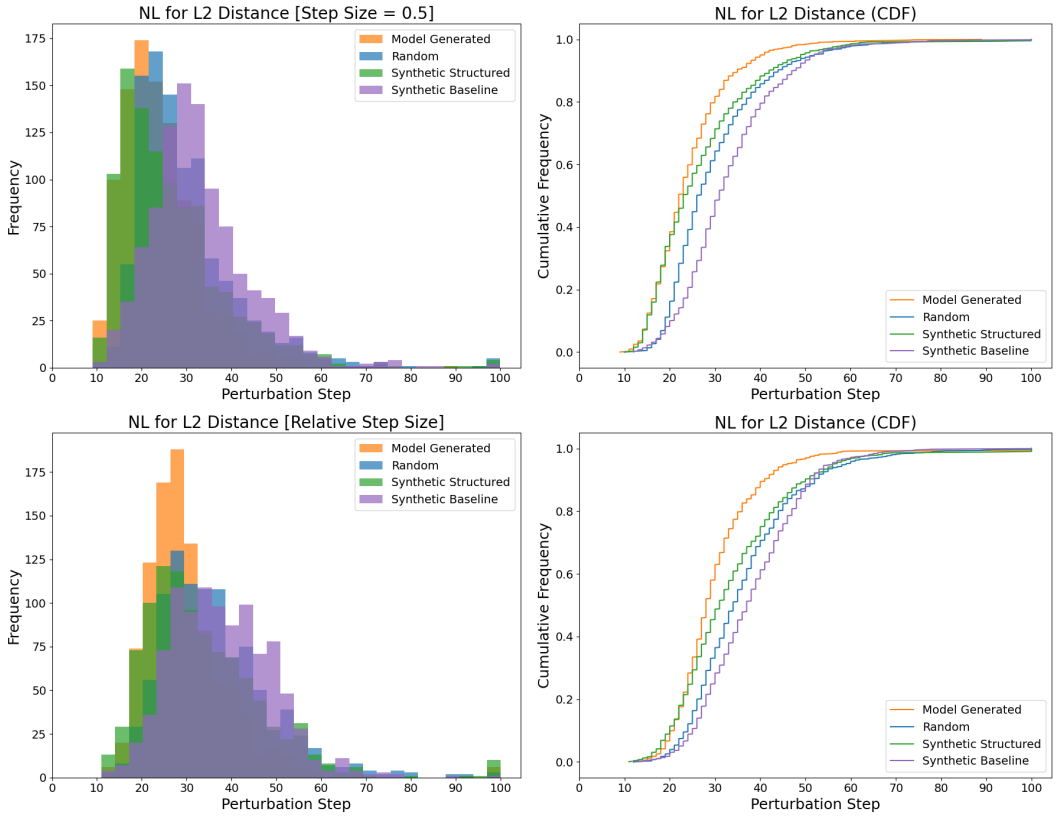

Figure B.2: The distributions of the NL steps for perturbations with absolute step size (top) and relative step size (bottom) towards model-generated (orange), random (blue), synthetic-baseline (purple), and synthetic-structured (green) activations. The left column shows the counts of NL steps occurring in different bins along the length of the perturbation, and the right column shows the cumulative frequency for the same. We find that synthetic-structured activations and random activations behave more like model-generated activations than synthetic-baseline activations do.

Non-Linear (NL) Step Distribution Statistics

Activation Type	Absolute Step Size			Relative Step Size		
	Mean	Std dev	KS	Mean	Std dev	KS
Model Generated	24.17	8.87	0.00	29.98	9.95	0.00
Random	29.80	11.72	0.22	36.33	12.33	0.27
Synthetic Baseline	32.90	11.25	0.40	37.91	10.96	0.37
Synthetic Structured	26.72	12.25	0.11	33.69	13.44	0.17

Table B.2: In terms of NL step distributions, we find that synthetic-structured activations perform better than random activations, but synthetic-baseline activations do not. This table contains the mean, standard deviation and KS statistic for NL step distributions for all the perturbations we perform. The KS statistic is measured against perturbations towards model-generated activations, with a lower value meaning higher similarity.

C KL Divergence

While previous works have predominantly used KL divergence as a measure of sensitivity, our analysis revealed potential limitations of this approach. We observed that KL divergence produces a step-function-like curve even when linear perturbations are performed at the final layer of the model right before the unembedding. This behavior suggests that the step-function shape might be an artifact of the KL divergence metric itself (or possibly due to softmax), rather than a true representation

of activation plateaus. The logarithmic nature of KL divergence may amplify differences as they become larger, leading to a more pronounced blowup region and a flatter initial plateau region.

With the mentioned caveats in mind, we perform perturbations at Layer 1 and observe their effect on KL divergence of the logits distribution instead of L2 distance at Layer 11. Figure C.1 illustrates the MS step distribution for KL divergence across different activation types. KL divergence blowups are more localized in the relative step size setup than L2 distance blowups, suggesting that the model's output distribution is more robust to noise than the model's final layer activations, only blowing up when more than 40% of the base activation has been replaced. Similar to the results for L2 distance, we find that perturbations towards synthetic-structured activations are more similar to perturbations towards model-generated activations than synthetic-baseline activations are. The difference between synthetic-structured and synthetic-baseline activations is more pronounced for KL divergence than L2 distance.

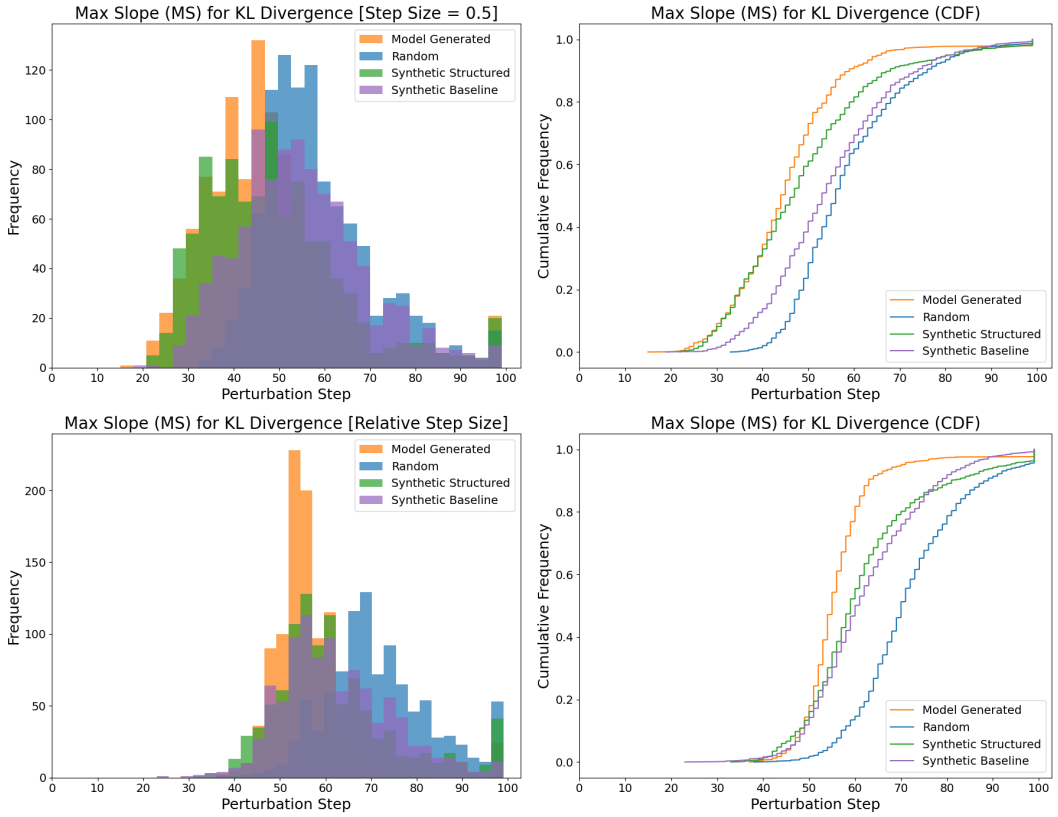

Figure C.1: The distributions of the MS steps for KL divergence of next-token prediction probabilities for perturbations with absolute step size (top) and relative step size (bottom) towards model-generated (orange), random (blue), synthetic-baseline (purple), and synthetic-structured (green) activations. The left column shows the counts of MS steps occurring in different bins along the length of the perturbation, and the right column shows the cumulative frequency for the same. We find that our results for KL divergence are similar to those for L2 distance.

Max Slope (MS) Step Distribution Statistics for KL divergence

Activation Type	Absolute Step Size			Relative Step Size		
	Mean	Std dev	KS	Mean	Std dev	KS
Model Generated	45.51	12.89	0.00	56.34	9.08	0.00
Random	58.54	**12.33**	0.47	71.83	**11.75**	0.69
Synthetic Baseline	54.79	14.02	0.32	62.41	12.05	0.32
Synthetic Structured	**48.79**	15.64	**0.13**	**61.86**	13.51	**0.26**

Table C.1: We find that our results for KL divergence of next-token prediction probabilities are similar to those for L2 distance at Layer 11. This table contains the mean, standard deviation and KS statistic for MS step distributions for all the perturbations we perform. The KS statistic is measured against perturbations towards model-generated activations, with a lower value meaning higher similarity.

D Isolating the effect of SAE reconstruction error

We denote the reconstruction of an activation A with `SAE(A) = decode(encode(A))`. To isolate the effect of SAE reconstruction error on the blowup location, we examine perturbations towards a reconstruction of a model-generated target activation `SAE(T)`. We compare these to perturbations towards model-generated activations and find that they are very similar, with blowups occurring slightly later for perturbations towards SAE reconstructions (Figure D.1, Table D.1). We also find that reconstructions of model-generated activations also have plateaus around them. This shows that the majority of the difference in our synthetic activations comes from the heuristics we use to select latents, and not the SAE reconstruction error.

This similarity suggests that SAE reconstructions behave like model-generated activations for the most part, and that the reconstruction error causes a small systematic shift in the blowup location. This points to some information loss that causes the model to respond slightly less to perturbations towards SAE reconstruction, which is relevant for interpreting experiments that use SAE latents.

Max Slope (MS) Step Distribution Statistics for SAE reconstructions

Activation Type	Absolute Step Size			Relative Step Size		
	Mean	Std dev	KS	Mean	Std dev	KS
Model Generated	41.11	10.40	0.00	51.60	7.82	0.00
Random	52.49	**10.21**	0.45	65.01	11.19	0.61
SAE Reconstruction	**41.49**	11.34	**0.02**	**53.34**	**8.39**	**0.11**

Table D.1: We find that perturbations towards model-generated activations are almost identical to perturbations towards their SAE reconstructions. This table contains the mean, standard deviation and KS statistic for MS step distributions for all the perturbations we perform. The KS statistic is measured against perturbations towards model-generated activations, with a lower value meaning higher similarity.

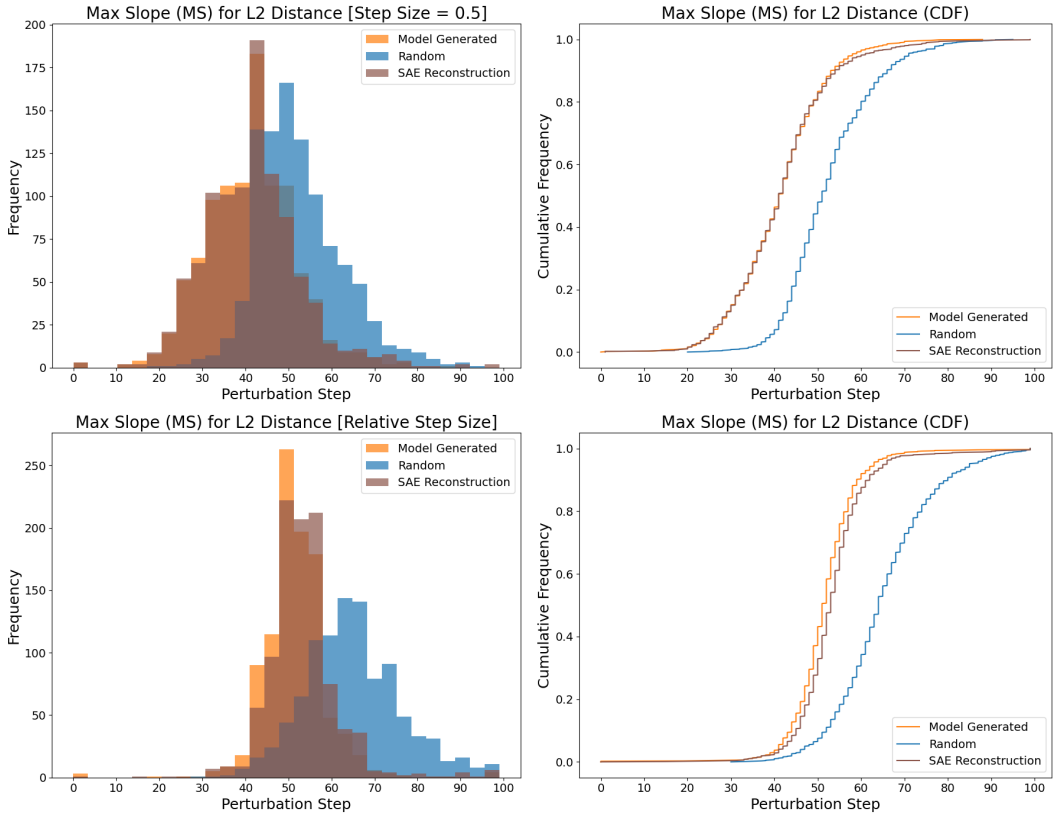

Figure D.1: The distributions of the MS steps for perturbations with absolute step size (top) and relative step size (bottom) towards random activations (blue), model-generated activations (orange), and their SAE reconstructions (brown). The left column shows the counts of MS steps occurring in different bins along the length of the perturbation, and the right column shows the cumulative frequency for the same. We find that perturbations towards model-generated activations and perturbations towards their SAE reconstructions are almost identical.

E Properties of SAE latents in model activations

We observe that model-generated activations with a low SAE reconstruction error contain approximately 21 active SAE latents on average (Figure E.1 left). The distribution is narrow around the mean and falls off very rapidly. The top latent represents around 49% of the total latent activation norm average (Figure E.1 right). The norm falls off rapidly thereafter, with the second top latent representing only around 10% on average. The distribution flattens out afterwards where latter ranks have similar contribution to the norm.

Additionally, we find that model-generated activations are made up of SAE latents that have cosine similarity to one another of approximately 0.29 on average (Figure E.2 left), with a distinct peak at 0. SAE latents primarily have positive cosine similarity to the top SAE latent, with mean cosine similarity of 0.18 (Figure E.2 right) and with a more pronounced peak at 0.

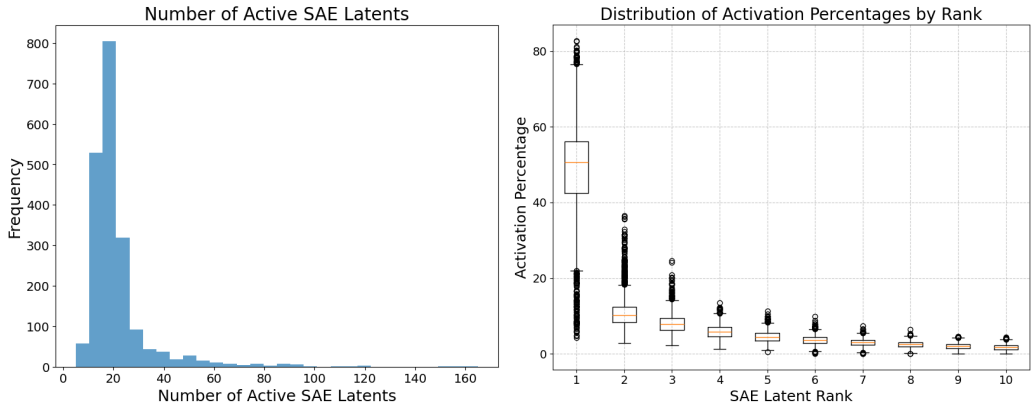

Figure E.1: The distribution of the total number of active SAE latents per activation (left) and the distribution of the percentage of the latent activation norm represented by the top 10 active latents (right) aggregated over 2000 activations.

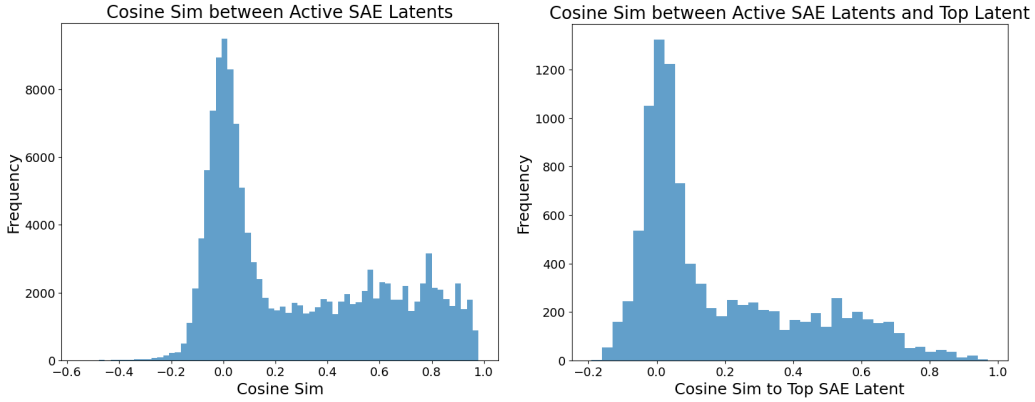

Figure E.2: The distribution of cosine similarities between all active SAE latents per activation (left) and distribution of cosine similarities that active SAE latents have with the top SAE latent (right) aggregated over 2000 activations.