Evaluating Synthetic Activations composed of SAE Latents in GPT-2

Anonymous Author(s)
Affiliation
Address
email

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 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]. To explain this, the theory of *superposition* suggests that neural networks represent more features than they have dimensions, with features linearly represented as directions in activation space [Elhage et al., 2022, Bricken et al., 2023]. However, the claim that all features are represented as directions remains speculative [Engels et al., 2024, Smith, 2024, Olah, 2024].

Sparse Auto-Encoders (SAEs) have become increasingly popular for decomposing a model's residual stream into more interpretable latents Sharkey et al. [2022], Bricken et al. [2023], Cunningham et al. [2023]. As reliance on SAEs grows, it is crucial to verify that they accurately capture model-used abstractions.

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. These are characterized by changes in the L2 distance of model activations at the final layer in response to early layer perturbations. Activation plateaus indicate model robustness to small amounts of noise, while directional sensitivity refers to the model's varied response to perturbations in different directions. Importantly, activation plateaus are present around model-generated activations (real) but not around random points sampled from the Gaussian approximation of the distribution of model-generated activations (random).

[*]Equal contribution

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

In this paper, we generate synthetic activations composed of SAE latents and test if they behave like real activations. We investigate whether arbitrary combinations of SAE latents, “bags of SAE latents”, can produce activations resembling real ones, and explore the role of latent sparsity and cosine similarity in this process. Our key contributions include:

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

Several studies have explored model responses to residual stream perturbations:

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] showed that SAE reconstructions cause larger KL divergence shifts in model outputs compared to equidistant random vectors when substituted for original activations. 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] investigated SAE reconstruction errors and end-to-end SAE latents, focusing on individual latent directions. Our work differs by studying compositions of SAE latents.

Lindsey [2024] examined the effects of ablating and dampening SAE latents on model performance. In our study, we focus on composing synthetic activations and studying SAE latent properties.

4 Method

We adapt the experimental settings from Heimersheim and Mendel [2024] to test whether synthetic activations composed of SAE latents exhibit behaviors similar to model-generated (real) activations. This approach allows us to study key relationships between SAE latents for generating in-distribution synthetic activations. Section 4.1 outlines the directional sensitivity experiment methodology, Sections 4.2 and 4.3 describe the activation types tested, and Section 4.4 details the activation plateau experiment.

4.1 Perturbation Setup

We perturb activations at the last token position in layer 1 (`blocks.1.hook_resid_pre`). The unperturbed base activation A is perturbed towards a direction D:

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

where n is the step number (0 to 100), and D is the normalized difference between base and target activations. We use a step size of 0.5, making perturbation norms comparable to typical activation norms ($\simeq 56$).

We measure L2 distance between original and perturbed activations after the final layer (`blocks.11.hook_resid_post`), preferring it over KL divergence for clearer activation plateau structure (KL divergence results in Appendix C).

To locate blowups, we use the maximum slope (MS) step of the L2 distance curve (Figure 1; we discuss alternative metrics in Appendix B).

We use `GPT2-small` [Radford et al., 2019] for our experiments, running inference on random `10`-token prompts from `OpenWebText` [Gokaslan and Cohen, 2019][*]. Model-generated activations are collected from Layer 1 (`blocks.1.hook_resid_pre`). We employ `GPT2-small` SAEs [Bloom, 2024], `sae-lens` [Bloom and Chanin, 2024], and `TransformerLens` [Nanda and Bloom, 2022] for experiments and synthetic activation generation.

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 and obtain base activations, and perturb each base activation in two directions:

- **Model-generated (real):** Towards a randomly selected activation produced by the model.
- **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. Both baselines have similar base-target distances (mean $\simeq$ 40, corresponding to step 80). The average cosine similarity between model-generated activations (w.r.t. SAE decoder bias) is $\simeq 0.42$.

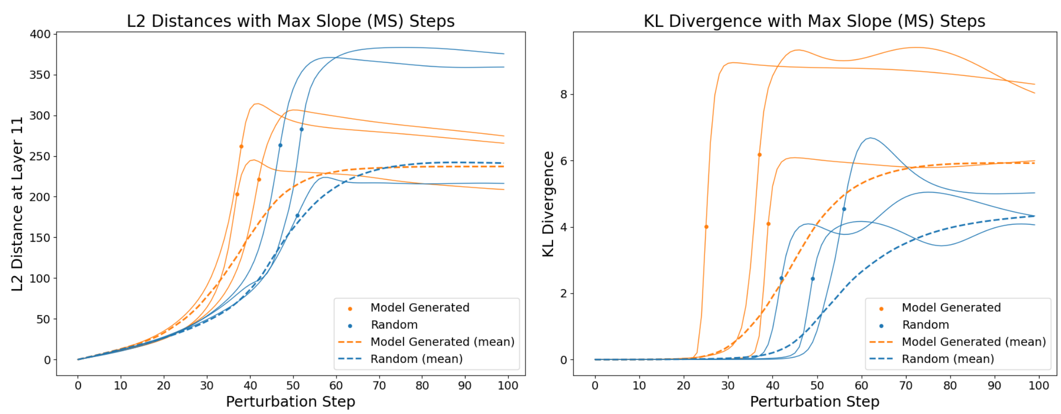

Figure 1: L2 distance (left) and KL divergence (right) between perturbed and unperturbed models for perturbations towards model-generated (orange) and random (blue) activations. X-axis: 100 perturbation steps of 0.5 size each. Dots: maximum slope steps. Dashed lines: average over 1000 perturbations. Initial linear part: activation plateau; sharp rise: blowup.

[*]Tokenized dataset link anonymized for review

4.3 Synthetic Activations

We construct synthetic activations using three methods, each incorporating different levels of SAE latent information:

Synthetic-random: Randomly selects SAE latents, assigning them base activation's latent activations. We present results for this activation type in Appendix A.

Synthetic-baseline: Accounts for SAE latent sparsities and activations ("bag of SAE latents"):

1. Encode base activation to obtain active latents and their activations.
2. Replace each active latent with a random one from the 10 most similarly-sparse latents and assign the same activation as the original latent.
3. Decode to obtain synthetic-baseline activation.

Synthetic-structured: Additionally captures geometric properties of real activations' SAE latents:

1. Encode base activation, identifying active latents and their activations. Define `top_base` as the latent with highest activation.
2. Create a list of 100 non-dead SAE latents with the most similar sparsity to `top_base`. Out of the 100 selected latents, select one 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`.
3. 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`.
4. For each remaining active latent in the base activation:
    (a) Calculate its cosine similarity (`l_top_cos_sim`) with `top_base`.
    (b) Select a latent (`l_synth`) that has cosine similarity with `top_synth` equal to `l_top_cos_sim`.
    (c) Assign `l_synth` a latent activation value equal to that of `l_base`.
5. 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 per synthetic activation type, as described in Section 4.1.

4.4 Activation Plateaus

To test if synthetic activations exhibit activation plateaus like real activations, we use the following approach:

1. Initiate perturbations from four base activation types: model-generated, synthetic-baseline, synthetic-structured, and random (as described in Section 4.2).
2. Perturb all base types towards random activations (as described in Section 4.2).
3. Record the activation plateau (AP) step where L2 distance at Layer 11 crosses 20, indicating plateau flatness.

We perform 1000 perturbations per base type, collecting AP size distributions. Larger AP steps indicate flatter activation plateaus.

5 Results

Synthetic activations behave differently from real and random activations across our two experiments, suggesting directional sensitivity and activation plateaus point to different properties of SAE latents in real activations (details in Appendix E). We primarily focus on studying directional sensitivity, though we also include our findings regarding activation plateaus below.

5.1 Directional Sensitivity

We perturb real activations towards different activation types and study the model's sensitivity. Figure 2 and Table 1 show the distributions and statistics of max slope (MS) steps for L2 distance across perturbation types.

Perturbations towards real activations cause earlier and more localized blowups compared to random activations, indicating higher model sensitivity. Synthetic-baseline activations, while not fully replicating real activation behavior, outperform random activations. We use the Kolmogorov-Smirnov statistic [Smirnov, 1948] to measure distribution similarities.

Synthetic-structured activations more closely resemble model-generated activations than synthetic-baseline or random activations do. (Figure 2, Table 1). This suggests that relationships between SAE latents are important, and that model-generated activations are not approximated well by "bags of SAE latents". Synthetic-random activations perform worse than synthetic-baseline, validating our choice of the latter as a stronger baseline (details in Appendix A).

To account for varying distances between base and target activations, we also perform perturbations with relative step size (Appendix A). This reduces the gap between synthetic-structured and synthetic-baseline performance, as synthetic-baseline activations are typically further from base activations and thus cause later blowups. It also decreases performance of synthetic-structured.

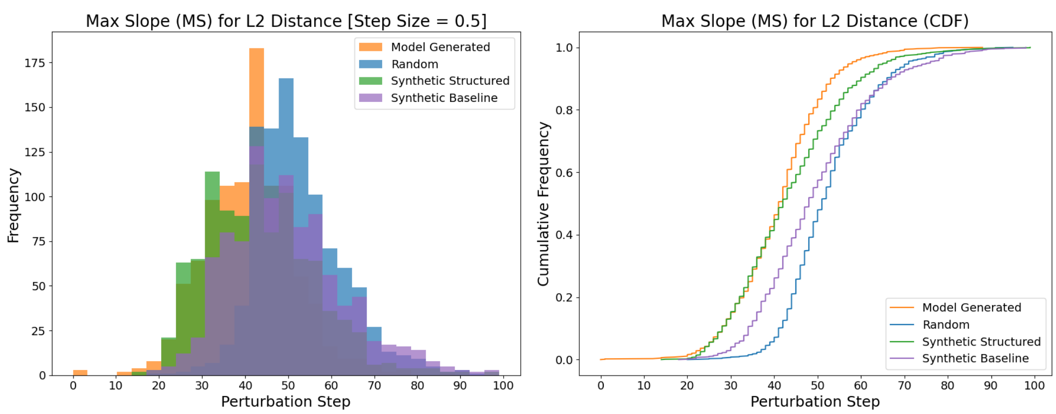

Figure 2: Distribution of MS steps for perturbations towards model-generated (orange), random (blue), synthetic-baseline (purple), and synthetic-structured (green) activations. Left: histogram; Right: cumulative frequency. Synthetic-structured perturbations more closely resemble model-generated ones compared to synthetic-baseline.

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: Mean, standard deviation, and KS statistic of MS step distributions for perturbations with fixed step size. KS statistic measured against model-generated activations (lower values indicate higher similarity). Synthetic-structured activations most closely resemble model-generated ones.

5.2 Activation Plateaus

Starting from a base of different activation types, we towards random directions to assess their activation plateaus. Figure 3 shows the distributions of AP steps for L2 distance across activation types.

Model-generated activations display pronounced activation plateaus that are not present around in random activations. We find that neither synthetic-baseline or synthetic-structured activations show such plateaus, providing further evidence against the "bag of SAE latents" approach but also showing that our synthetic-structure activations do not capture all relevant properties of real activations.

The SAE reconstruction error minimally contributes to the discrepancy between synthetic and model-generated activations, we test this in Appendix D.

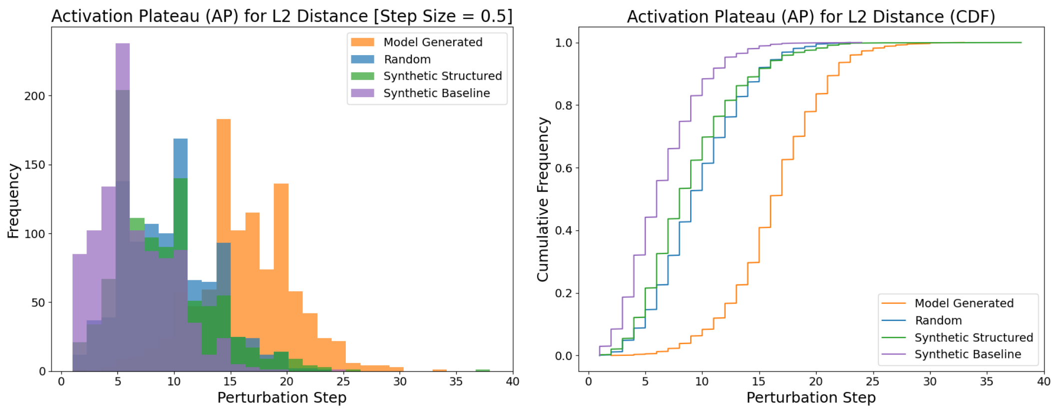

Figure 3: Distribution of activation plateau (AP) steps for perturbations from various activation types towards random activations. Left: histogram; Right: cumulative frequency. Model-generated activations (orange) show flattest plateaus; synthetic-baseline (purple) the steepest. Synthetic-structured (green) and random (blue) activations show similar plateau characteristics.

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.

Synthetic activations don't fully match the cosine similarity distribution of SAE latents in model-generated activations (Appendix E).

Our method 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.

L2 distance curves' variability may affect our MS metric's effectiveness, as it assumes curve smoothness (Figure 1). More robust metrics could yield clearer results (Appendix B).

Our study is limited to one early layer of GPT2-small and the final token position. Further research across different layers, models, SAEs, and context lengths is needed to establish broader applicability and generalizability of our findings.

7 Conclusion

Our findings provide additional evidence that GPT-2 is more sensitive to perturbations towards model-generated activations than random directions, and that model-generated activations are not merely "bags of SAE latents". Leveraging statistical and geometric properties of SAE latents allows us to create synthetic-structured activations more similar to model-generated ones, indicating that they capture important properties of SAE latents. However, these lack characteristic plateaus of model-generated activations, suggesting additional SAE latent properties influence model computation.

This presents exciting avenues for future work on model sensitivity to perturbations: developing improved synthetic activation construction methods; investigating thresholds for model response to latent activation changes; examining model sensitivity to perturbations using interpretable SAE latents and contextual information; and analyzing latent ablation-based perturbations to identify key contributors to blowups.

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. Boundaries of stable regions in activation space of llms become sharper with more compute, September 2024.

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.

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:

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

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.

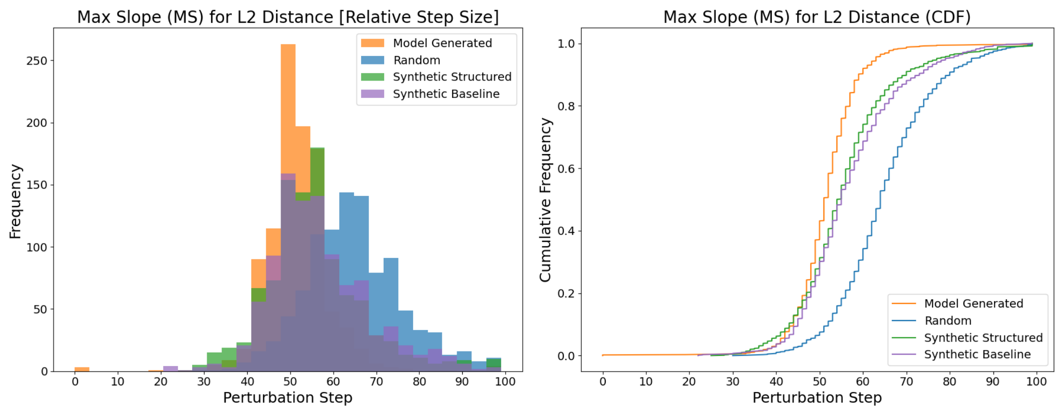

Figure A.1: The distributions of the max slope (MS) steps for perturbations with relative step size towards model-generated (orange), random (blue), synthetic-baseline (purple), and synthetic-structured (green) activations. The left panel shows the counts of MS steps occurring in different bins along 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.

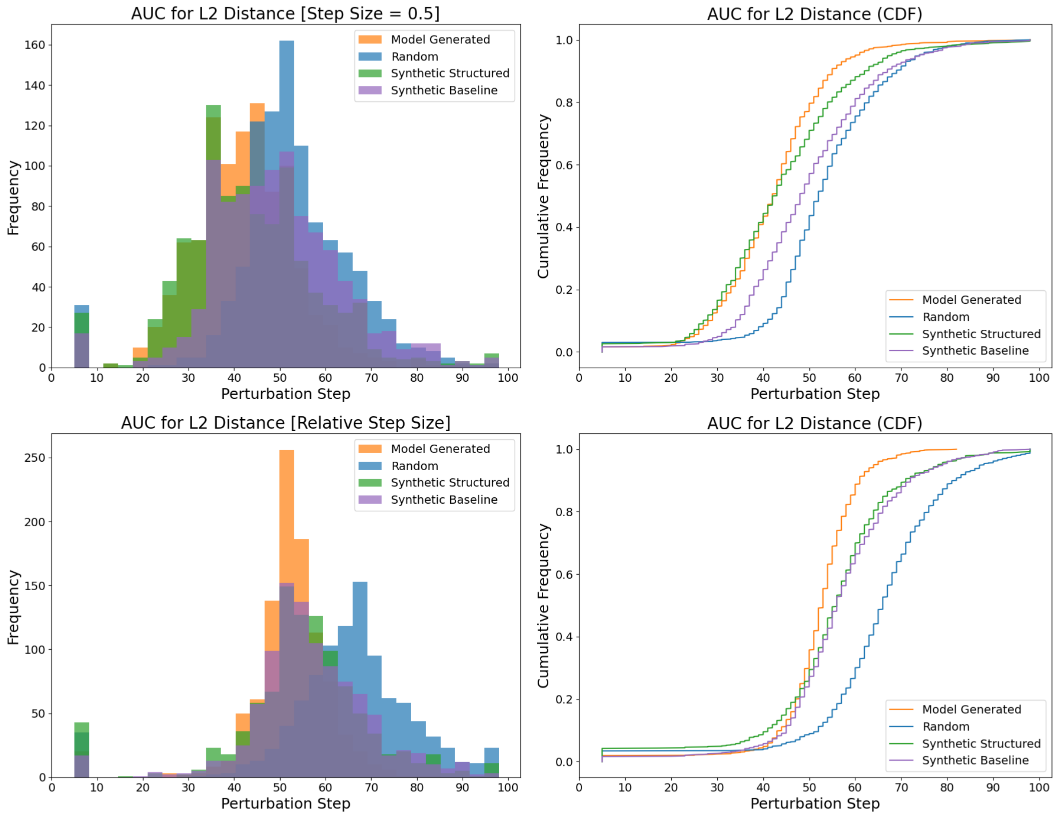

Figure B.1: The distributions of the AUC steps for perturbations with absolute step size (top) and relative step size (bottom) towards model-generated (orange), random (blue), synthetic-baseline (purple), and synthetic-structured (green) activations. The left column shows the counts of AUC steps occurring 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.

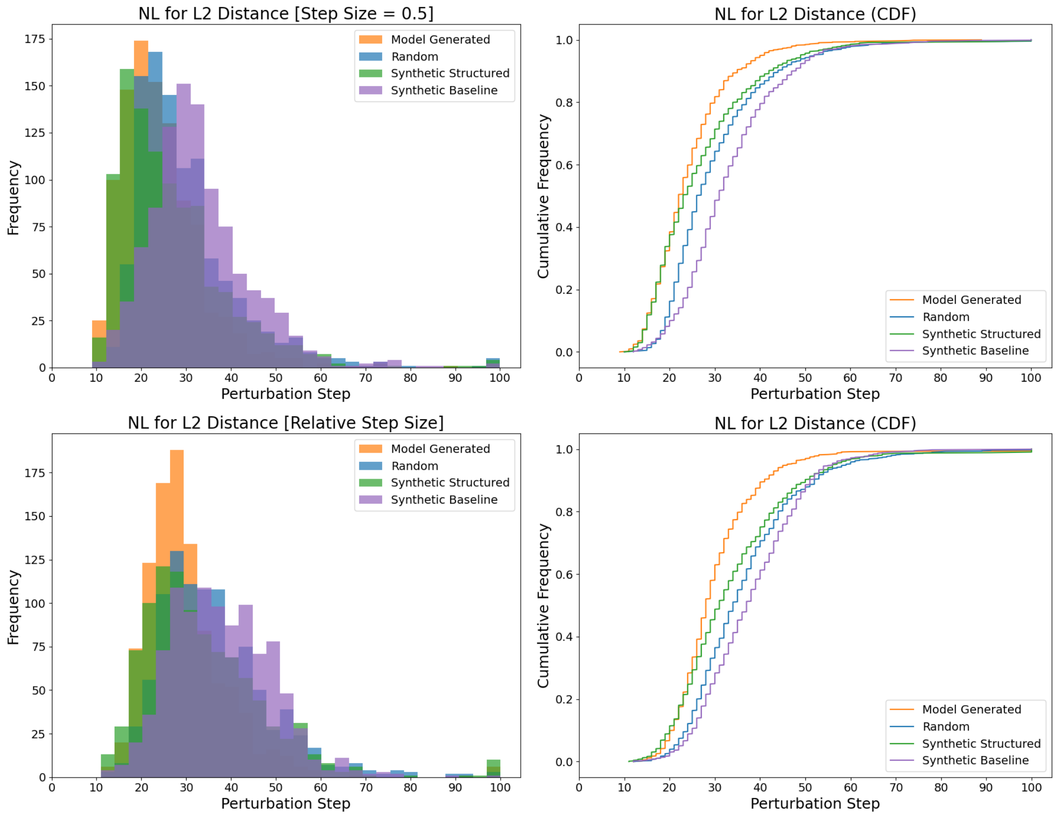

Figure B.2: The distributions of the NL steps for perturbations with absolute step size (top) and relative step size (bottom) towards model-generated (orange), random (blue), synthetic-baseline (purple), and synthetic-structured (green) activations. The left column shows the counts of NL steps occurring 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.

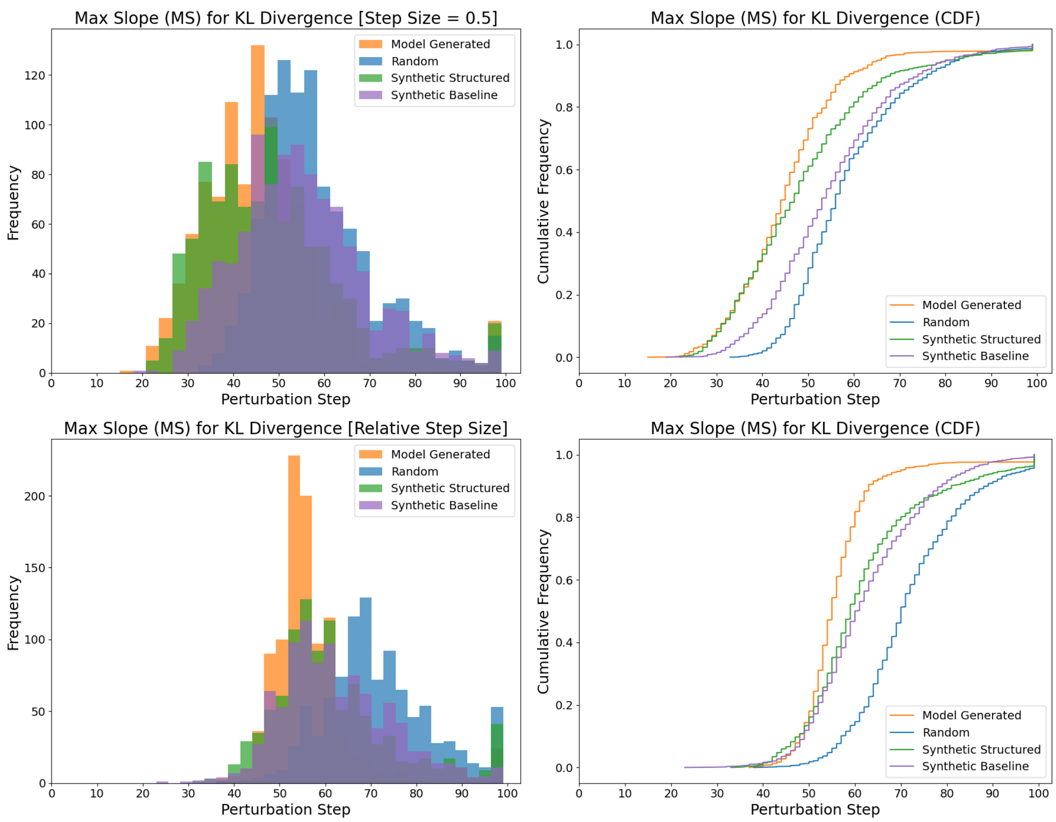

Figure C.1: The distributions of the MS steps for KL divergence of next-token prediction probabilities for perturbations with absolute step size (top) and relative step size (bottom) towards model-generated (orange), random (blue), synthetic-baseline (purple), and synthetic-structured (green) activations. 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.

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**

Max Slope (MS) Step Distribution Statistics for KL divergence

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.

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**

Max Slope (MS) Step Distribution Statistics for SAE reconstructions

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.

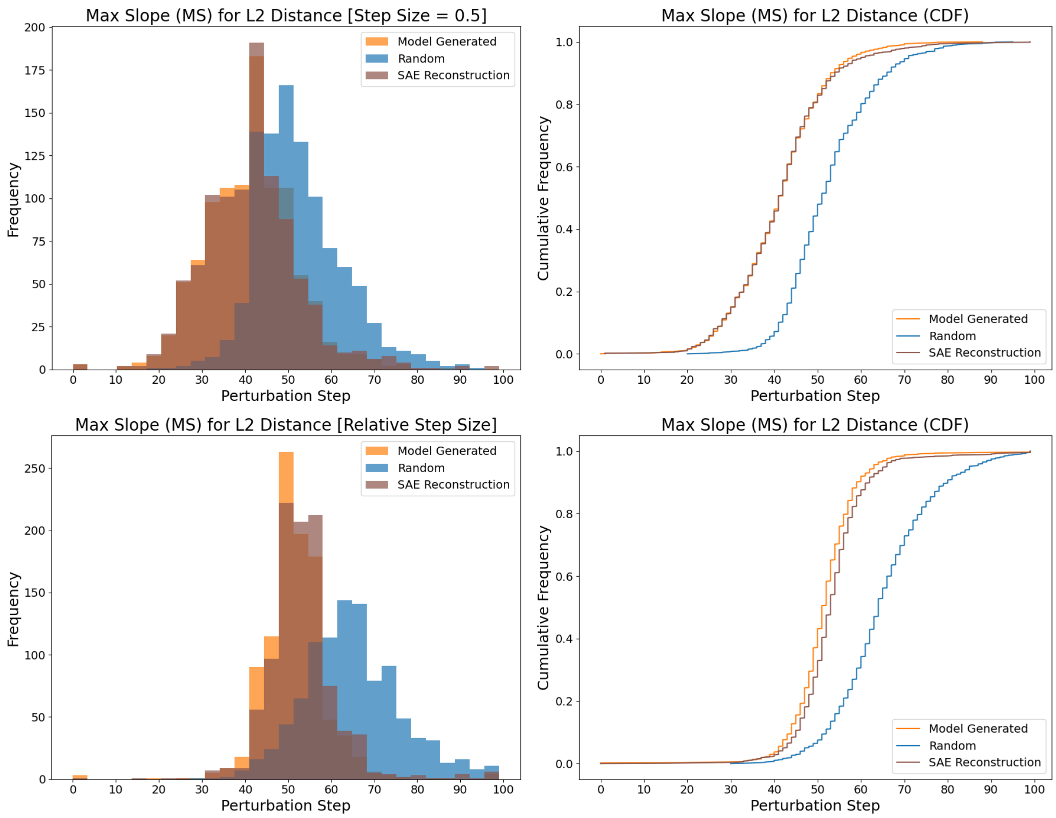

Figure D.1: The distributions of the MS steps for perturbations with absolute step size (top) and relative step size (bottom) towards random activations (blue), model-generated activations (orange), and their SAE reconstructions (brown). The left column shows the counts of MS steps occurring in different 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.

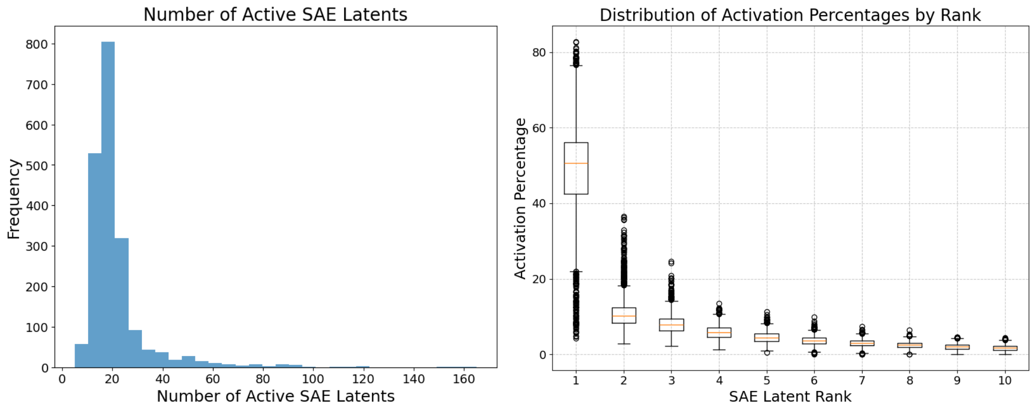

Figure E.1: The distribution of the total number of active SAE latents per activation (left) and the distribution of the percentage of the latent activation norm represented by the top 10 active latents (right) aggregated over 2000 activations.

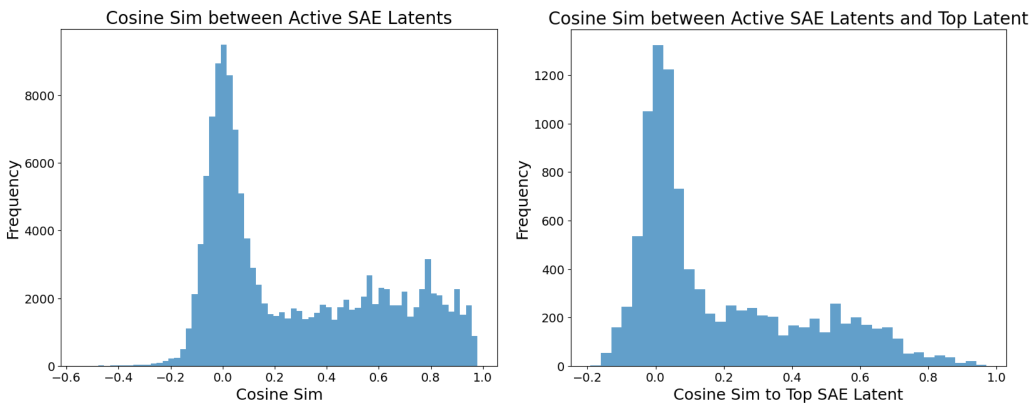

Figure E.2: The distribution of cosine similarities between all active SAE latents per activation (left) and distribution of cosine similarities that active SAE latents have with the top SAE latent (right) aggregated over 2000 activations.