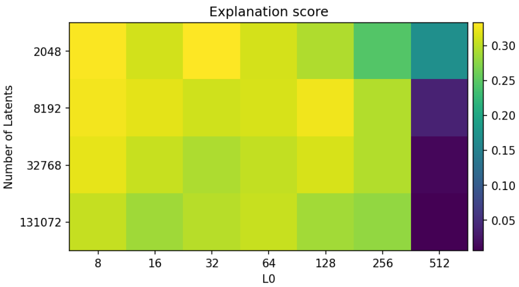


Figure 21: Explanation scores for GPT-2 small autoencoders of different $n$ and $k$, evaluated on 400 randomly chosen latents per autoencoder. It is hard to read off trends, but the explanation score is able to somewhat detect the dense solutions region.

### E.6 Recurring dense features in GPT-2 small

We manually examined the densest latents across various GPT-2 small layer 8 autoencoders, trained in different ways (e.g. differing numbers of total latents).

The two densest latents are always the same feature: the latent simply activates more and more later in the context ($\sim$ 40% active), and one that activates more and more earlier in the context excluding the first position ($\sim$35% active). Both features look like they want to activate more than they do, with TopK probably preventing it from activating with lower values.

The third densest latent is always a first-token-position feature ($\sim$30% active), which has a modal activation value in a narrow range between 14.6-14.8. Most of its activation values are significantly smaller values, at tokens after the first position; the large value is always at the first token. These smaller values appear uninterpretable; we conjecture these are simply interference with the first position direction. (Sometimes there are two of these latents, the second with smaller activation values.)

Finally, there is a recurring "repetition" feature that is $\sim$20% dense. Its top activations are mostly highly repetitive sequences, such as series of dates, chapter indices, numbers, punctuations, repeated exact phrases, or other repetitive things such as Chess PGN notation. However, like the first-token-position latents, random activations of this latent are typically appear unrelated and uninterpretable.

Often in the top ten densest latents, we find opposing latents, which have decoder cosine similarity close to $-1$. In particular, the first-token-position feature and the repetition latent both seems to always have an opposite latent. The less dense of the two opposite latents always seems to appear uninterpretable. We conjecture that these are symptoms of optimization failure - the opposite latents cancel out spurious activations in the denser latent.

### E.7 Clustering latents

[Elhage et al., 2022] discuss how underlying features may lie in distinct sub-spaces. If such sub-spaces exists, we hypothesize that the set of latent encoding vectors $W \in \mathbb{R}^{n \times d}$ can be written as a block-diagonal matrix $W' = PWR$, where $P \in \mathbb{R}^{n \times n}$ is a permutation matrix, and $R \in \mathbb{R}^{d \times d}$ is orthogonal. We can then use the singular vector decomposition (SVD) to write $W = U \Sigma V^\top$ and $W' = U' \Sigma' V'^\top$, noting that $U'$ is also block diagonal. Finally, we write $W = P^\top U' \Sigma' V'^\top R^\top = U \Sigma V^\top$, and because the SVD is unique up to a column permutation $P'$, we get $U = P^\top U' P'$. In other words, if $W$ is block-diagonal in some unknown basis, $U$ is also block diagonal up to a permutation of rows and columns.

To find a good permutation of rows, we sorted the rows of $U$ based on how similarly they project on all elements of the singular vector basis. Specifically, we normalized each row to unit norm $\tilde{U_i} = U_i / ||U_i||$ and considered the pairwise euclidean distances $d_{i,j} = ||\tilde{U_i}^2 - \tilde{U_j}^2||$. These pairwise distances were then reduced to a single dimension with a UMAP algorithm [McInnes et al., 2018]. The obtained 1-dimensional embedding was then used to order the projections $\tilde{U_i}^2$ (Figure 22a),

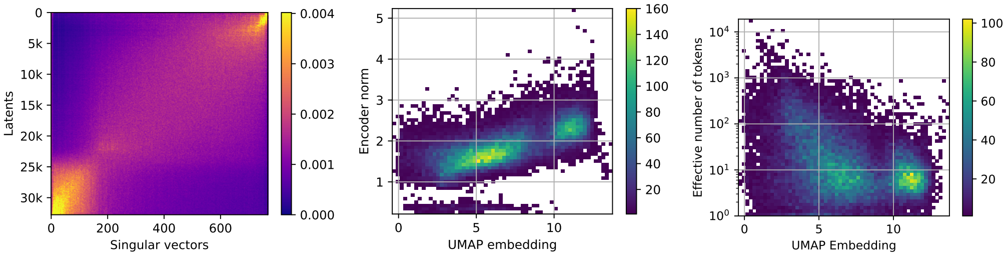


Figure 22: The residual stream seems composed of two separate sub-spaces. About 25% of latents mostly project on a sub-space using 25% of dimensions. These latents tend to have larger encoder norm, and to activate on a smaller number of vocabulary tokens. The remaining 75% of latents mostly project on the remaining 75% of dimensions, and can activate on a larger number of vocabulary tokens.

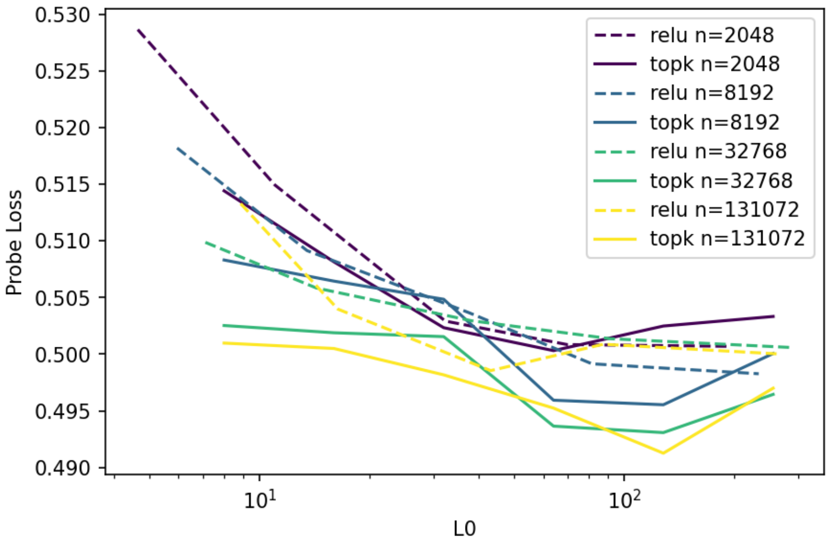


Figure 23: TopK beats ReLU not only on the sparsity-MSE frontier, but also on the sparsity-probe loss frontier. (lower is better)

which reveals two fuzzily separated sub-spaces. These two sub-spaces use respectively about 25% and 75% of the dimensions of the entire vector space.

Interestingly, ordering the columns by singular values is fairly consistent with these two sub-spaces. One reason for this result might by that latents projecting to the first sub-space have different encoder norms than latents projecting to the second sub-space (Figure 22b). This difference in norm can significantly guide the SVD to separate these two sub-spaces.

To further interpret these two sub-spaces, we manually looked at latents from each cluster. We found that latents from the smaller cluster tend to activate on relatively non-diverse vocabulary tokens. To quantify this insight, we first estimated $A_{i,v}$, the average squared activation of latent $i$ on vocabulary token $v$. Then, we normalized the vectors $A_i$ to sum to one, $\tilde{A}_{i,v} = A_{i,v} / \sum_w A_{i,w}$ and computed the effective number of token $m_i = \exp(\sum_v \tilde{A}_{i,v} \log(\tilde{A}_{i,v}))$. The effective number of token is a continuous metric with values in $[1, n_{\text{vocab}}]$, and it is equal to $k$ when a latent activates equally on $k$ vocabulary tokens. With this metric, we confirmed quantitatively (Figure 22c) that latents from the smaller cluster all activate on relatively low numbers of vocabulary tokens (less than $100$), whereas latents from the larger cluster sometimes activate on a larger numbers of vocabulary tokens (up to $1000$).

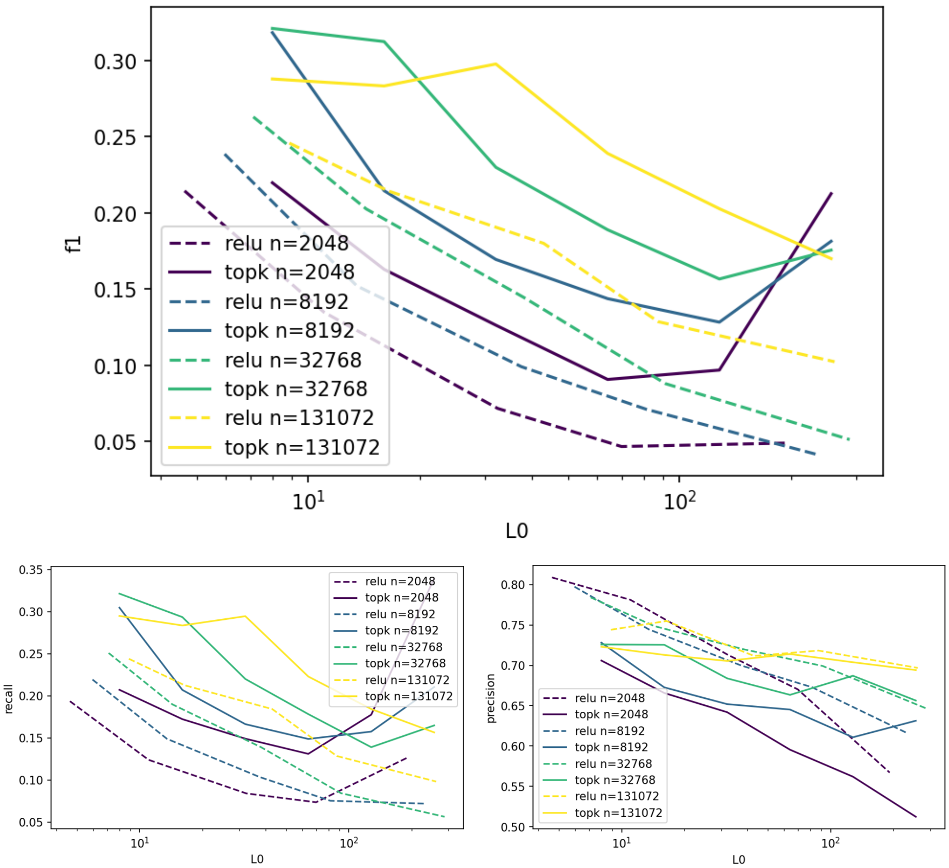
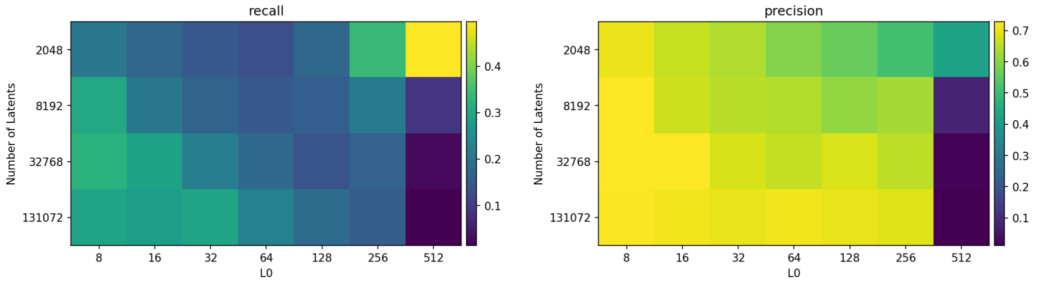


Figure 24: TopK beats ReLU on N2G F1 score. Its N2G explanations have noticeably higher recall, but worse precision. (higher is better)



(a) Recall of N2G explanations
$P(\text{n2g} > 0|\text{act} > 0)$

(b) Precision of N2G explanations
$P(\text{act} > 0|\text{n2g} > 0)$

Figure 25: Neuron2graph precision and recall. The average autoencoder latent is generally easier to explain as $k$ decreases and $n$ increases. However, $n = 2048, k = 512$ latents are easy to explain since many latents activate extremely densely (see Section E.5).

# F Miscellaneous small results

## F.1 Impact of different locations

In a sweep across locations in GPT-2 small, we found that the optimal learning rate varies with layer and location type (MLP delta, attention delta, MLP post, attention post), but was within a factor of two.