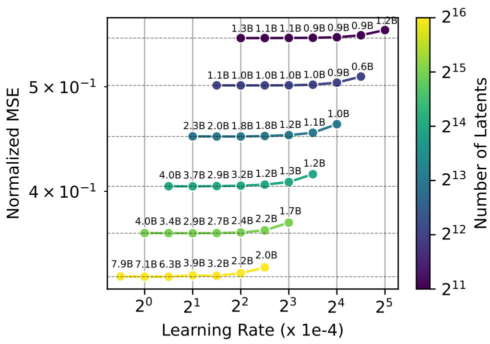


Figure 3: Varying the learning rate jointly with the number of latents. Number of tokens to convergence shown above each point.

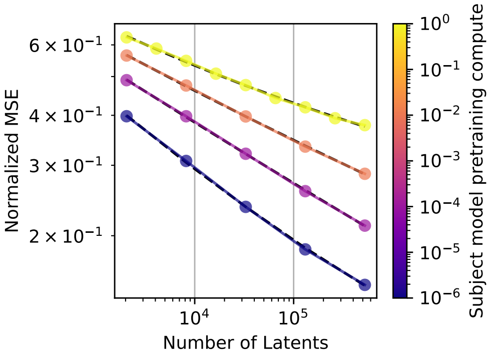


Figure 4: Larger subject models in the GPT-4 family require more latents to get to the same MSE ($k = 32$).

3 Scaling laws

3.1 Number of latents

Due to the broad capabilities of frontier models such as GPT-4, we hypothesize that faithfully representing model state will require large numbers of sparse features. We consider two primary approaches to choose autoencoder size and token budget:

3.1.1 Training to compute-MSE frontier ($L(C)$)

Firstly, following Lindsey et al. [2024], we train autoencoders to the optimal MSE given the available compute, disregarding convergence. This method was introduced for pre-training language models [Kaplan et al., 2020, Hoffmann et al., 2022]. We find that MSE follows a power law $L(C)$ of compute, though the smallest models are off trend (Figure 1).

However, latents are the important artifact of training (not reconstruction predictions), whereas for language models we typically care only about token predictions. Comparing MSE across different n is thus not a fair comparison — the latents have a looser information bottleneck with larger n, so lower MSE is more easily achieved. Thus, this approach is arguably unprincipled for autoencoder training.

3.1.2 Training to convergence ($L(N)$)

We also look at training autoencoders to convergence (within some ϵ). This gives a bound on the best possible reconstruction achievable by our training method if we disregard compute efficiency. In practice, we would ideally train to some intermediate token budget between $L(N)$ and $L(C)$.

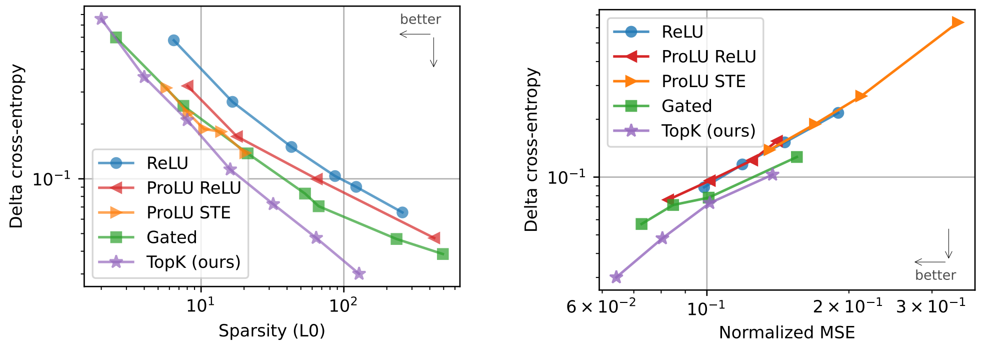
We find that the largest learning rate that converges scales with $1/\sqrt{n}$ (Figure 3). We also find that the optimal learning rate for $L(N)$ is about four times smaller than the optimal learning rate for $L(C)$.

We find that the number of tokens to convergence increases as approximately $\Theta(n^{0.6})$ for GPT-2 small and $\Theta(n^{0.65})$ for GPT-4 (Figure 11). This must break at some point – if token budget continues to increase sublinearly, the number of tokens each latent receives gradient signal on would approach zero.[8]

3.1.3 Irreducible loss

Scaling laws sometimes include an irreducible loss term e, such that $y = \alpha x^{\beta} + e$ [Henighan et al., 2020]. We find that including an irreducible loss term substantially improves the quality of our fits for both $L(C)$ and $L(N)$.

[8] One slight complication is that in the infinite width limit, TopK autoencoders with our initialization scheme are actually optimal at init using our init scheme (Section A.1), so this allows for an exponent very slightly less than 1; however, this happens very slowly with n so is unlikely to be a major factor at realistic scales.



(a) For a fixed number of latents ($n = 2^{17} = 131072$), the downstream-loss/sparsity trade-off is better for TopK autoencoders than for other activation functions.

(b) For a fixed sparsity level ($L_0 = 128$), a given MSE level leads to a lower downstream-loss for TopK autoencoders than for other activation functions.

Figure 5: Comparison between TopK and other activation functions on downstream loss. Comparisons done for GPT-2 small, see Figure 13 for GPT-4.

It was initially not clear to us that there should be a nonzero irreducible loss. One possibility is that there are other kinds of structures in the activations. In the extreme case, unstructured noise in the activations is substantially harder to model and would have an exponent close to zero (Appendix G). Existence of some unstructured noise would explain a bend in the power law.

3.1.4 Jointly fitting sparsity ($L(N, K)$)

We find that MSE follows a joint scaling law along the number of latents n and the sparsity level k (Figure 1b). Because reconstruction becomes trivial as k approaches d_{model}, this scaling law only holds for the small k regime. Our joint scaling law fit on GPT-4 autoencoders is:

$$L(n, k) = \exp(\alpha + \beta_k \log(k) + \beta_n \log(n) + \gamma \log(k) \log(n)) + \exp(\zeta + \eta \log(k)) \tag{3}$$

with $\alpha = -0.50$, $\beta_k = 0.26$, $\beta_n = -0.017$, $\gamma = -0.042$, $\zeta = -1.32$, and $\eta = -0.085$. We can see that γ is negative, which means that the scaling law $L(N)$ gets steeper as k increases. η is negative too, which means that the irreducible loss decreases with k.

3.2 Subject model size $L_s(N)$

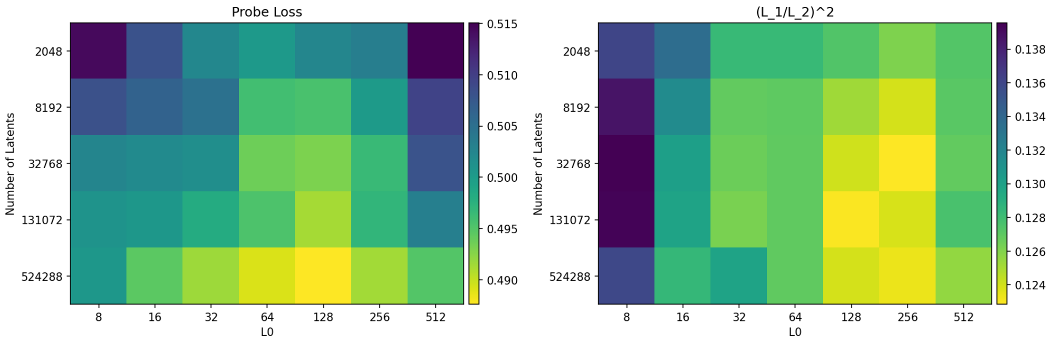
Since language models are likely to keep growing in size, we would also like to understand how sparse autoencoders scale as the subject models get larger. We find that if we hold k constant, larger subject models require larger autoencoders to achieve the same MSE, and the exponent is worse (Figure 4).

4 Evaluation

We demonstrated in Section 3 that our larger autoencoders scale well in terms of MSE and sparsity (see also a comparison of activation functions in Section 5.2). However, the end goal of autoencoders is not to improve the sparsity-reconstruction frontier (which degenerates in the limit[9]), but rather to find features useful for applications, such as mechanistic interpretability. Therefore, we measure autoencoder quality with the following metrics:

1. **Downstream loss**: How good is the language model loss if the residual stream latent is replaced with the autoencoder reconstruction of that latent? (Section 4.1)
2. **Probe loss**: Do autoencoders recover features that we believe they might have? (Section 4.2)

[9] Improving the reconstruction-sparsity frontier is not always strictly better. An infinitely wide maximally sparse ($k = 1$) autoencoder can perfectly reconstruct by assigning latents densely in $\mathbb{R}^d$, while being completely structure-less and uninteresting.



(a) Probe loss

(b) Logit diff sparsity

Figure 6: The probe loss and logit diff metrics as a function of number of total latents n and active latents k, for GPT-2 small autoencoders. More total latents (higher n) generally improves all metrics (yellow = better). Both metrics are worse at $\mathrm{L}_0 = 512$, a regime in which solutions are dense (see Section E.5).

3. **Explainability**: Are there simple explanations that are both necessary and sufficient for the activation of the autoencoder latent? (Section 4.3)
4. **Ablation sparsity**: Does ablating individual latents have a sparse effect on downstream logits? (Section 4.5)

These metrics provide evidence that autoencoders generally get better when the number of total latents increases. The impact of the number of active latents L_0 is more complicated. Increasing L_0 makes explanations based on token patterns worse, but makes probe loss and ablation sparsity better. All of these trends also break when L_0 gets close to d_{model}, a regime in which latents also become quite dense (see Section E.5 for detailed discussion).

4.1 Downstream loss

An autoencoder with non-zero reconstruction error may not succeed at modeling the features most relevant for behavior [Braun et al., 2024]. To measure whether we model features relevant to language modeling, we follow prior work [Bills et al., 2023, Cunningham et al., 2023, Bricken et al., 2023, Braun et al., 2024] and consider downstream Kullback-Leibler (KL) divergence and cross-entropy loss.[10] In both cases, we test an autoencoder by replacing the residual stream by the reconstructed value during the forward pass, and seeing how it affects downstream predictions. We find that k-sparse autoencoders improve more on downstream loss than on MSE over prior methods (Figure 5a). We also find that MSE has a clean power law relationship with both KL divergence, and difference of cross entropy loss (Figure 5b), when keeping sparsity L_0 fixed and only varying autoencoder size. Note that while this trend is clean for our trained autoencoders, we can observe instances where it breaks such as when modulating k at test time (see Section 5.3).

One additional issue is that raw loss numbers alone are difficult to interpret—we would like to know how good it is in an absolute sense. Prior work [Bricken et al., 2023, Rajamanoharan et al., 2024] use the loss of ablating activations to zero as a baseline and report the fraction of loss recovered from that baseline. However, because ablating the residual stream to zero causes very high downstream loss, this means that even very poorly explaining the behavior can result in high scores.[11]

Instead, we believe a more natural metric is to consider the relative amount of pretraining compute needed to train a language model of comparable downstream loss. For example, when our 16 million latent autoencoder is substituted into GPT-4, we get a language modeling loss corresponding to 10% of the pretraining compute of GPT-4.

[10] Because a perfect reconstruction would lead to a non-zero cross-entropy loss, we actually consider the difference to the perfect-autoencoder cross-entropy ("delta cross-entropy").

[11] For completeness, the zero-ablation fidelity metric of our 16M autoencoder is 98.2%.