APPROACH 1: CREATING MODELS WITHOUT SUPERPOSITION

It's actually quite easy to get rid of superposition in the toy models described in this paper, albeit at the cost of a higher loss. Simply apply at L1 regularization term to the hidden layer activations (i.e. add $\lambda||h||_1$ to the loss). This actually has a nice interpretation in terms of killing features below a certain importance threshold, especially if they're not basis aligned. Generalizing this to real neural networks isn't trivial, but we expect it can be done.

However, it seems likely that models are significantly benefitting from superposition. Roughly, the sparser features are, the more features can be squeezed in per neuron. And many features in language models seem very sparse! For example, language models know about individuals with only modest public presences, such as several of the authors of this paper. Presumably we only occur with frequency significantly less than one in a million tokens. As a result, it may be the case that superposition effectively makes models much bigger.

All of this paints a picture where getting rid of superposition may be fairly achievable, but doing so will have a large performance cost. For a model with a fixed number of neurons, superposition helps – potentially a lot.

But this is only true if the constraint is thought of in terms of neurons. That is, a superposition model with n neurons likely has the same performance as a significantly larger monosemantic model with kn neurons. But neurons aren't the fundamental constraint: flops are. In the most common model architectures, flops and neurons have a strict correspondence, but this doesn't have to be the case and it's much less clear that superposition is optimal in the broader space of possibilities.

One family of models which change the flop-neuron relationship are Mixture of Experts (MoE) models (*see review* [39]). The intuition is that most neurons are for specialized circumstances and don't need to activate most of the time. For example, German-specific neurons don't need to activate on French text. Harry Potter neurons don't need to activate on scientific papers. So MoE models organize neurons into blocks or experts, which only activate a small fraction of the time. This effectively allows the model to have k times more neurons for a similar flop budget, given the constraint that only $1/k$ of the neurons activate in a given example and that they must activate in a block. Put another way, MoE models can recover neuron sparsity as free flops, as long as the sparsity is organized in certain ways.

It's unclear how far this can be pushed, especially given difficult engineering constraints. But there's an obvious lower bound, which is likely too optimistic but is interesting to think about: what if models only expended flops on neuron activations, and recovered the compute of all non-activating neurons? In this world, it seems unlikely that superposition would be optimal: you could always split a polysemantic neuron into dedicated neurons for each feature with the same cost, except for the cases where there would have been interference that hurt the model anyways. Our preliminary investigations comparing various types of superposition in terms of "loss reduction per activation frequency" seem to suggest that superposition is not optimal on these terms, although it asymptotically becomes as good as dedicated feature dimensions. Another way to think of this is that superposition exploits a gap between the sparsity of neurons and the sparsity of the underlying features; MoE eats that same gap, and so we should expect MoE models to have less superposition.

To be clear, MoE models are already well studied, and we don't think this changes the capabilities case for them. (If anything, superposition offers a theory for why MoE models have not proven more effective for capabilities when the case for them seems so initially compelling!) But if one's goal is to create competitive models that don't have superposition, MoE models become interesting to think about. We don't necessarily think that they specifically are the right path forward – our goal here has been to use them as an example of why we think it remains plausible there may be ways to build competitive superposition-free models.

APPROACH 2: FINDING AN OVERCOMPLETE BASIS

The opposite strategy of creating a superposition-free model is to take a regular model, which has superposition, and find an overcomplete basis describing how features are embedded after the fact.