- **Is there a statistical test for catching superposition?**
- **How can we control whether superposition and polysemanticity occur?** Put another way, can we change the phase diagram such that features don't fall into the superposition regime? Pragmatically, this seems like the most important question. L1 regularization of activations, adversarial training, and changing the activation function all seem promising.
- **Are there any models of superposition which have a closed-form solution?** Saxe et al. [26] demonstrate that it's possible to create nice closed-form solutions for linear neural networks. We made some progress towards this for the $n = 2; m = 1$ ReLU output model (and Tom McGrath makes further progress in his comment), but it would be nice to solve this more generally.
- **How realistic are these toy models?** To what extent do they capture the important properties of real models with respect to superposition? How can we tell?
- **Can we estimate the feature importance curve or feature sparsity curve of real models?** If one takes our toy models seriously, the most important properties for understanding the problem are the feature importance and sparsity curves. Is there a way we can estimate them for real models? (Likely, this would involve training models of varying sizes or amounts of regularization, observing the loss and neuron sparsities, and trying to infer something.)
- **Should we expect superposition to go away if we just scale enough?** What assumptions about the feature importance curve and sparsity would need to be true for that to be the case? Alternatively, should we expect superposition to remain a constant fraction of represented features, or even to increase as we scale?
- **Are we measuring the maximally principled things?** For example, what is the most principled definition of superposition / polysemanticity?
- **How important are polysemantic neurons?** If X% of the model is interpretable neurons and 1-X% are polysemantic, how much should we believe we understand from understanding the x% interpretable neurons? (See also the "feature packing principle" suggested above.)
- **How many features should we expect to be stored in superposition?** This was briefly discussed in the previous section. It seems like results from compressed sensing should be able to give us useful upper-bounds, but it would be nice to have a clearer understanding – and perhaps tighter bounds!
- **Does the apparent phase change we observe in features/neurons have any connection to phase changes in compressed sensing?**
- **How does superposition relate to non-robust features?** An interesting paper by Gabriel Goh (archive.org backup) explores features in a linear model in terms of the principal components of the data. It focuses on a trade off between "usefulness" and "robustness" in the principal component features, but it seems like one could also relate it to the interpretability of features. How much would this perspective change if one believed the superposition hypothesis – could it be that the useful, non-robust features are an artifact of superposition?
- **To what extent can neural networks "do useful computation" on features in superposition?** Is the absolute value problem representative of computation in superposition generally, or idiosyncratic? What class of computation is amenable to being performed in superposition? Does it require a sparse structure to the computation?
- **How does superposition change if features are not independent?** Can superposition pack features more efficiently if they are anti-correlated?
- **Can models effectively use nonlinear representations?** We suspect models will tend not to use them, but further experimentation could provide good evidence. See the appendix on nonlinear compression. For example investigating the representations used by autoencoders with multi-layer encoders and decoders with really small bottlenecks on random uncorrelated data.

# Related Work

### INTERPRETABLE FEATURES

Our work is inspired by research exploring the features that naturally occur in neural networks. Many models form at least some interpretable features. Word embeddings have semantic directions (*see* [8]). There is evidence of interpretable neurons in RNNs (*e.g.* [11, 12]), convolutional neural networks (*see generally e.g.* [13, 14, 41, 19]; *individual neuron families* [6, 18]), and in some limited cases, transformer language models (*see detailed discussion in our previous paper*). However this work has also found many "polysemantic" neurons which are *not* interpretable as a single concept [21].

### SUPERPOSITION

We're aware of two separate origins of the idea of superposition in neural networks. The first is the superposition hypothesis explored in this paper. The existence of polysemantic neurons (described in the previous section) led to the superposition hypothesis as one of the most plausible seeming explanations [1]. This hypothesis is a kind of "feature level" superposition.

Separately, Cheung *et al.* [7] explore what one might describe as "model level" superposition: can neural network parameters represent multiple completely independent models? Their investigation is motivated by catastrophic forgetting, but seems quite related to the questions investigated in this paper.

### DISENTANGLEMENT

The goal of learning *disentangled representations* arises from Bengio *et al.*'s influential position paper on representation learning [5]: "we would like our representations to *disentangle the factors of variation*… to learn representations that separate the various explanatory sources." Since then, a literature has developed motivated by this goal, tending to focus on creating genderantive models which separate out major factors of variation in their latent spaces.

Concretely, disentanglement research often explores whether one can train a VAE or GAN where basis dimensions correspond to the major features one might use to describe the problem (e.g. rotation, lighting, gender… as relevant). In the language of this paper, the goal is to impose a strong privileged basis on the latent space of a generative model, which are often totally rotationally invariant by default. Early work often focused on semi-supervised approaches where the features were known in advance, but fully unsupervised approaches started to develop around 2016 [42, 43, 44]

How does superposition relate to disentanglement? Although our investigation was motivated primarily by different examples, we see no reason to think that superposition doesn't also occur in the latent spaces of generative models. If so, it may be that superposition is a major reason why disentanglement is difficult. Superposition may allow generative models to be much more effective than they would otherwise be without. Put another way, disentanglement often assumes a small number of important latent variables explain the data. There are clearly examples of such variables, like the orientation of objects – but what if a large number of sparse, rare, individually unimportant features are collectively very important? Superposition would be the natural way for models to represent this.[22]

### COMPRESSED SENSING

The toy problems we consider are quite similar to the problems considered in the field of compressed sensing, which is also known as compressive sensing and sparse recovery. However, there are some important differences:

- Compressed sensing recovers vectors by solving an optimization problem using general techniques, while our toy model must use a neural network layer. Compressed sensing algorithms are, in principle, much more powerful than our toy model
- Compressed sensing works using the number of non-zero entries as the measure of sparsity, while we use the probability that each dimension is zero as the sparsity. These are not wholly unrelated: concentration of measure implies that our vectors have a bounded number of non-zero entries with high probability.
- Compressed sensing requires that the embedding matrix (usually called the measurement matrix) have a certain "incoherent" structure [45] such as the restricted isometry property [25] or nullspace property [46]. Our toy model learns the embedding matrix, and will often simply ignore many input dimensions to make others easier to recover.
- Features in our toy model have different "importances", which means the model will often prefer to be able to recover "important" features more accurately, at the cost of not being able to recover "less important" features at all.

In general, our toy model is solving a similar problem using *less powerful* than compressed sensing algorithms, especially because the computational model is so much more restricted (to just a single linear transformation and a non-linearity) compared to the arbitrary computation that might be used by a compressed sensing algorithm.

As a result, compressed sensing lower bounds—which give lower bounds on the dimension of the embedding such that recovery is still possible—can be interpreted as giving an upper bound on the amount of superposition in our toy model. In particular, in various compressed sensing settings, one can recover an $n$-dimensional $k$-sparse vector from an $m$ dimensional projection if and only if $m = \Omega(k \log(n/k))$ [47, 48, 49]. While the connection is not entirely straightforward, we apply one such result to the toy model in the appendix.

At first, this bound appears to allow a number of features that is exponential in $m$ to be packed into the $m$-dimensional embedding space. However, in our setting, the integer $k$ for which all vectors have at most $k$ non-zero entries is determined by the fixed density parameter $S$ as $k = O((1-S)n)$. As a result, our bound is actually $m = \Omega(-n(1-S)\log(1-S))$. Therefore, the number of features is linear in $m$ but modulated by the sparsity.[23] This is good news if we are hoping to eliminate superposition as a phenomenon! However, these bounds also allow for the amount of superposition to increase dramatically with sparsity – hopefully this is an artifact of the techniques in the proofs and not an inherent barrier to reducing or eliminating superposition.

A striking parallel between our toy model and compressed sensing is the existence of *phase changes*.[24] In compressed sensing, if one considers a two-dimensional space defined by the sparsity and dimensionality of the vectors, there are sharp phase changes where the vector can almost surely be recovered in one regime and almost surely not in the other [50, 51]. It isn't immediately obvious how to connect these phase changes in compressed sensing – which apply to recovery of the entire vector, rather than one particular component – to the phase changes we observe in features and neurons. But the parallel is suspicious.

Another interesting line of work has tried to build useful sparse recovery algorithms using neural networks [52, 53, 54]. While we find it useful for analysis purposes to view the toy model as a sparse recovery algorithm, so that we may apply sparse recovery lower bounds, we do not expect that the toy model is useful for the problem of sparse recovery. However, there may be an exciting opportunity to relate our understanding of the phenomenon of superposition to these and other techniques.