THE PERSIAN RUG: SOLVING TOY MODELS OF SUPERPOSITION USING LARGE-SCALE SYMMETRIES

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a complete mechanistic description of the algorithm learned by a minimal non-linear sparse data autoencoder in the limit of large input dimension. The model, originally presented in Elhage et al. (2022), compresses sparse data vectors through a linear layer and decompresses using another linear layer followed by a ReLU activation. We notice that when the data is permutation symmetric (no input feature is privileged) large models reliably learn an algorithm that is sensitive to individual weights only through their large-scale statistics. For these models, the loss function becomes analytically tractable. Using this understanding, we give the explicit scalings of the loss at high sparsity, and show that the model is near-optimal among recently proposed architectures. In particular, changing or adding to the activation function any elementwise or filtering operation can at best improve the model's performance by a constant factor. Finally, we forward-engineer a model with the requisite symmetries and show that its loss precisely matches that of the trained models. Unlike the trained model weights, the low randomness in the artificial weights results in miraculous fractal structures resembling a Persian rug, to which the algorithm is oblivious. Our work contributes to neural network interpretability by introducing techniques for understanding the structure of autoencoders.

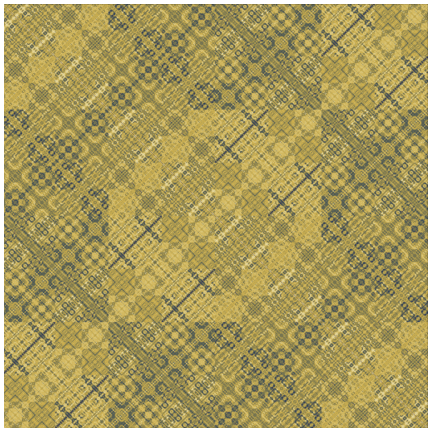

Figure 1: The Persian rug, an artificial set of weights matching trained model performance.

1 INTRODUCTION

Large language model capabilities and applications have recently proliferated. As these systems advance and are given more control over basic societal functions, it becomes imperative to ensure their reliability with absolute certainty. Mechanistic interpretability aims to achieve this by obtaining a concrete weight-level understanding of the algorithms learned and employed by these models. A major impediment to this program has been the difficulty of interpreting intermediate activations. This is due to the phenomena of superposition, in which a model takes advantage of sparsity in the input data to reuse the same neurons for multiple distinct features, obscuring their function. Finding

a systematic method to undo superposition and extract the fundamental features encoded by the network is a large and ongoing area of research Bricken et al. (2023); Cunningham et al. (2023); Gao et al. (2024); Engels et al. (2024).

Currently, the most popular method of dealing with superposition is dictionary learning with sparse autoencoders. In this method, the smaller space of neuron activations at a layer of interest is mapped to a larger feature space. The map is trained to encourage sparsity and often consists of an affine + ReLU network. This method has been applied to large language models revealing many strikingly interpretable features (e.g. corresponding to specific bugs in code, the golden gate bridge, and sycophancy), even allowing for a causal understanding of the model's reasoning in certain scenarios Marks et al. (2024).

The sparse decoding ability of the affine + ReLU map was recently studied in the foundational work Elhage et al. (2022), which introduced and studied a toy model of superposition. The model consisted of a compressing linear layer modeling the superposition[1] followed by a decompressing affine + ReLU layer, trained together to auto-encode sparse data. They showed that the network performs superposition by encoding individual feature vectors into nearly orthogonal vectors in the smaller space. The affine layer alone is unable to decode sparse input vectors sufficiently well to make use of superposition, but the addition of the ReLU makes it possible by screening out negative interference.

While Elhage et al. (2022) provides valuable empirical and theoretical insights into superposition, it does not obtain a strong enough description of the model algorithm to quantitatively characterize the algorithm's performance. Given the extensive use of the affine + ReLU map for decoding sparse data in practice, it is important to obtain a complete analytic understanding of the model behavior over a large parameter regime. As we will see, this will inform the design of better sparse autoencoder architectures.

In this work we obtain such an understanding by considering a particularly tractable regime of the Elhage et al. (2022) model: permutation symmetric data (no input feature is privileged in any way), and the thermodynamic limit (a large number of input features), while maintaining the full range of sparsity and compression ratio values. In this regime, the learned model weights are permutation symmetric on large scales, which sufficiently simplifies the form of the loss function to the point where it is analytically tractable, leaving only a small number of free parameters. We then forwards-engineer an artificial set of weights satisfying these symmetries and optimizing the remaining parameters, which achieves the same loss as a corresponding trained model, implying that trained models also implement the optimal permutation symmetric algorithm. The artificial set of weights resembles a Persian rug fig. 1, whose structure is a relic of the minimal randomness used in the construction, illustrating that the algorithm relies entirely on large-scale statistics that are insensitive to this structure. Finally, we derive the exact power-law scaling of the loss in the high-sparsity regime.

We expect our work to impact the field of neural network interpretability in multiple ways. First, our work provides a basic theoretical framework that we believe can be extended to other regimes of interest, such as structured correlations in input features, which may help predict scaling laws in the loss based on the data's correlations. Second, our work rules out a large class of performance improvement proposals for sparse autoencoders. Finally, our work provides an explicit example of a learned algorithm that is insensitive to microscopic structure in weights, which may be useful for knowing when not to analyze individual weights.

The paper is structured as follows. In section 2 we review the model and explain our training procedure. In section 3 we show empirically that large models display a "statistical" permutation symmetry. In section 4 we extract the algorithm by plugging the symmetry back into the loss, introduce the Persian rug model which optimizes the remaining parameters, show that large trained models achieve the same loss, and derive the loss behavior in the high sparsity limit. In section 5 we conclude and discuss related works.

[1] This is because, if good enough recovery is possible for most features, the pigeonhole principle tells us that at least some of the smaller space activations must encode information about multiple input features.

2 THE MODEL

We consider the following model for our non-linear autoencoder with matrix parameters $W_{\text{in}} \in \mathbb{R}^{n_d \times n_s}, W_{\text{out}} \in \mathbb{R}^{n_s \times n_d}, \mathbf{b} \in \mathbb{R}^{n_s}$,

$$f_{\text{nonlinear}}(\mathbf{x}) = \text{ReLU}(W_{\text{out}} W_{\text{in}} \mathbf{x} + \mathbf{b}). \tag{1}$$

W_{in} is an encoding matrix which converts a sparse activation vector $\mathbf{x}$ to a dense vector, while W_{out} perform the linear step of decoding. We also consider a simple model for the sparse data on which this autoencoder operates. We work with data that is permutation symmetric in the sense that $(x_1, ..., x_n)$ is equal in distribution to $(x_{\pi(1)}, x_{\pi(2)}, ..., x_{\pi(n)})$ for any permutation π. Each vector $\mathbf{x}$ is drawn i.i.d. during training, and each component is drawn i.i.d. with $x_i = c_i u_i$, where $c_i \sim \text{Bernoulli}(p)$ and $u_i \sim \text{Uniform}[0, 1]$ are independent variables. This ensures that $\mathbf{x}$ is sparse with typically only pn_s features turned on.

We train our toy models to minimize the expected L_2 reconstruction loss,

$$L(\mathbf{x}; W_{\text{out}}, W_{\text{in}}, \mathbf{b}) = n_s^{-1} \mathbb{E} ||\mathbf{x} - f_{\text{nonlinear}}(\mathbf{x})||_2^2. \tag{2}$$

It is known that for the linear model (eq. (1) without the ReLU), the optimal solution is closely related to principle component analysis (see, for example, Plaut (2018) and p. 563 or Bishop & Nasrabadi (2006)). In particular, the reconstruction loss decreases linearly in the hidden dimension n_d when all features are i.i.d. On the other hand, the model eq. (1) will have a much quicker reduction in loss, as will be described in section 3.1.

We train all models with a batch size of 1024 and the Adam optimizer to completion. That is training continues as long as the average loss over the past 200 batches is lower than the average loss over the 200 batches prior to that one. Our goal with training is to ensure that we have found an optimal model in the large-data limit to analyze the structure of the model itself. See also appendix G for more training details.

3 EMPIRICAL OBSERVATIONS

In this section, we present empirical observations of the trained models. We start by presenting a remarkable phenomenon this model exhibits in the high-sparsity regime: a dramatic decrease in loss as a function of the compression ratio. We then turn to a mechanistic interpretation of the weights which gives empirical evidence for the phenomena needed to understand the algorithm the model learns. These are manifestations of a partially preserved permutation symmetry of the sparse degrees of freedom.

3.1 FAST LOSS DROP

To gauge the performance of the model, we plot the loss (eq. (2)) as a function of the compression ratio n_d/n_s. In fig. 2 we plot the performance of the linear versus non-linear models for representative parameters. It is clear that the non-linear model outperforms the linear model up until near the $n_d/n_s \approx 1$ regime due to an immediate drop in the loss.The slope and duration of this initial fall is controlled by p. In particular, in the high-sparsity regime (p close to zero), the loss drops to zero entirely near the $n_d/n_s \approx 0$ regime. What is going on here? To explain this behavior, we analyze the algorithm the model encodes.

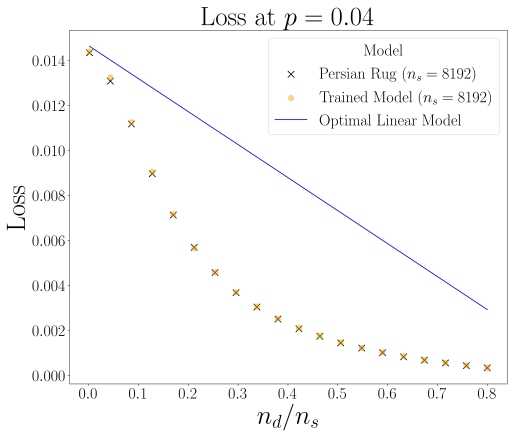

Figure 2: Loss curves of trained models, Persian rug models, and optimal linear models as a function of the compression ratio.

3.2 Statistical permutation symmetry

Rather than looking individually at the weights, it is helpful to look at the matrix $W = W_{\text{out}} W_{\text{in}}$, shown in fig. 3. As will be made precise in section 4, we are interested in the structure of each row since it is responsible for the corresponding feature output. Though the matrix is not permutation symmetric in detail, its statistics are sufficiently so as far as the algorithm is concerned. In particular, in the large n_s regime, the matrix is statistically permutation invariant in the following sense:

1. the diagonal elements become the same (fig. 4),
2. the bias elements become uniformly negative, which can be seen in the uniformity and slight blue shade in fig. 3 and is quantified in fig. 5,
3. the statistics of the off-diagonal terms in each row are sufficiently uniform for interference to become Gaussian (fig. 7), and finally
4. the mean and variance of each of these becomes the same across rows (fig. 6).

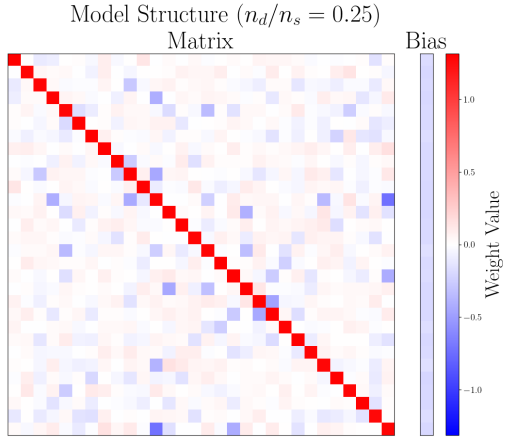

Figure 3: Plot of the first 30×30 W elements and the corresponding bias ($\mathbf{b}$) components, at $p = 4.5\%$ and ratio $n_s = 512$. The diagonal components are all at similar values of $1.29 \pm .01$ (one standard deviation) while the off-diagonal components are approximately mean-zero, appearing like noise. The bias elements are all negative around $-.18 \pm .01$. This statistical uniformity is a permutation symmetry across the sparse features.

Immediately on looking at the exemplar model from fig. 3 we see that the diagonals are large and uniform while the off-diagonal components are fluctuating and much smaller. This serves the function of routing each input to the corresponding output, except with some off-diagonal contributions which should be as small as possible.

Because the diagonal components appear uniform, we quantify this by looking at the fluctuation on a per-model basis. For each model, we compute the empirical mean and variance of the diagonal components and plot the latter in fig. 4. Initially, when $n_s = 128$ is small, we see a region with large fluctuations among the diagonal components, particularly when the p and n_d/n_s are both small. We hypothesize this occurs because for small models (and small feature probabilities) the contribution to any output from off-diagonal terms may fluctuate because of the small size of the matrix.

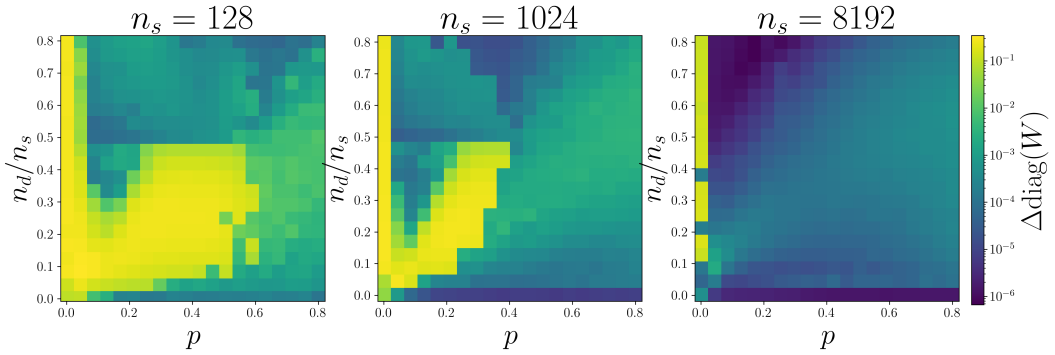

Figure 4: Permutation symmetry of diagonal values. We plot the mean-square fluctuation of the diagonal values corresponding to each model. Models are trained as a function of p and n_d/n_s. The emergence of symmetry as n_s grows (at all locations in the diagram) is a crucial element of the algorithm implemented by the autoencoders.

We see similar features among the entries of the bias in fig. 5 (this figure, along with the rest of the figures in this section, are similar to fig. 4 and so are placed in Appendix A for readability). The relatively large initial fluctuations generally decay as n_s grows larger. Similar features are present in fig. 4 and fig. 5, such as the relatively low fluctuation regions near $(p, n_d/n_s) \approx (.1, .5)$, because the bias and the diagonal entries are coupled; for example, when a diagonal element in a row is large the corresponding bias must be scaled up as well and vice versa.

For each of these algorithm-revealing features, we give three plots, each with increasing n_s, to show that the feature becomes apparent in the thermodynamic limit. Each plot is a heat map with the variable representing the feature plotted in color and spatial dimensions covering the entire parameter space of n_d/n_s and p.

We similarly see in fig. 6 that the off-diagonal terms concentrate as well. To be more precise, we look at the fluctuation of the total norm of the off-diagonal elements in each row in W across rows. The norm is a measure of the amount of signal arriving at a particular output from other inputs. A smaller contribution from other inputs means that each output has a higher signal-to-noise ratio as long as the diagonal term remains constant.

For a trained (deterministic) W and random x, the quantity $\sum_{j\neq i} W_{ij}x_j$ is a random variable which measures the extent to which the pre-activation reconstruction of x_i erroneously receives signal from other components x_j with $j \neq i$. Given that $\sum_{j\neq i} W_{ij}x_j$ is a sum of n_s independent random variables, it is natural to ask how well it can be approximated by a Gaussian in the large n_s regime. Clearly some conditions on W_{ij} and x_j will be needed; via the Berry-Esseen theorem for independent random variables, if the x_j have finite third moment, the quantity

$$\Lambda := \max_i \frac{\sum_{j\neq i} |W_{ij}|^3}{\left(\sum_{j\neq i} W_{ij}^2\right)^{3/2}}, \tag{3}$$

up to a constant factor that depends on the first three moments of x, is a measure of how far

$\sum_{j\neq i} W_{ij}x_j$ is from Gaussianity (see, for example, Petrov (1972)). In fig. 7, we plot Λ and show it tends to 0 as n_s grows larger.

3.3 Optimization of residual parameters

The statistical permutation symmetry places constraints on the possible values of W and $\mathbf{b}$. The constraint on $\mathbf{b}$ is straightforward: it is proportional to the all ones vector, i.e. there is a number b such that $\mathbf{b}_i = b$ for all i. As will be explained in section 4.1, the relevant degrees of freedom remaining in W are one number a equal to the diagonals, and σ characterizing the root mean square of the off diagonal rows. The precise values of the off diagonals can be thought of as irrelevant "microscopic information". Thus there are three relevant degrees of freedom remaining: b, a, and σ.

In section 4.2.2 we give a specific set of values for W_{in} and W_{out} via the "Persian rug" matrix, which have the statistical permutation symmetry in the $n_s \to \infty$ limit while also optimizing σ. The remaining parameters, a and b can be optimized numerically. In fig. 2 we compare the loss curve of this artificial model with that of a trained model, and see that they are essentially the same.

4 Extracting the Algorithm

In this section we give a precise explanation of the algorithm the model performs. We start with a qualitative description of why the statistical permutation symmetry gives a good auto-encoding algorithm when the remaining macroscopic degrees of freedom are optimized. We then find an artificial set of symmetric weights with optimized macroscopic parameters. We show that the trained models achieve the same performance as the artificial model, thus showing they are optimal even restricting to statistically symmetric solutions. Finally, we derive an explicit form of the loss scaling and argue that ReLU performs near optimally among element-wise or "selection" decoders.

4.1 Qualitative Description

A key simplification is to consider strategies as collections of low-rank affine maps rather than as the collection of weights directly. In other words consider the tuple $(W, \mathbf{b})$ where $W = W_{\text{out}}W_{\text{in}}$ to define the strategy. We must restrict to W with rank no more than n_d because it is the product of two low-rank matrices. Given any such W we may also find W_{in} and W_{out} of the appropriate shape (e.g. by finding the SVD), so the two representations are equivalent.

Let us now take a close look at the reconstruction error. Let us consider output i, which is given by

$$(f_{\text{nonlinear}}(\mathbf{x}))_i = \text{ReLU}(W_{ii}x_i + \sum_{j=1,j\neq i}^{n_s} W_{ij}x_j + \mathbf{b}_i) = \text{ReLU}\left(a(x_i + \nu_i) + \mathbf{b}_i\right) \tag{4}$$

where we used that all diagonal elements of W are equal to a and defining $\nu_i = a^{-1}\sum_{j=1,j\neq i}^{n_s} W_{ij}x_j$. At this stage we can already see that the individual off-diagonal elements are only important insofar as they characterize the Gaussian random variable ν_i. More specifically, because ν_i is completely characterized by its mean and variance, only the sum and sum of squares of the off-diagonal components matter. Because we have a bias term the mean of ν_i may be absorbed into it, so from now we assume that ν_i are all zero-mean. This simplification is why permutation symmetry emerges when n_s is large, and allows us to consider only the macroscopic variables introduced when we discuss the loss.

The reconstruction error is (dropping the i subscript, and substituting $\mathbf{b}_i = b$)

$$L = \mathbb{E}_{x,\nu}[(x - \text{ReLU}\left(a(x+\nu) + b\right))^2]. \tag{5}$$

We can further decompose this by taking the expectation value over whether x is on or off, so

$$L = (1-p)L_{\text{off}} + pL_{\text{on}}$$

where

$$L_{\text{off}} = \mathbb{E}_{\nu}[(\text{ReLU}\left(a\nu + b\right))^2], \text{ and}$$
$$L_{\text{on}} = \mathbb{E}_{u,\nu}[(u - \text{ReLU}\left(a(u+\nu) + b\right))^2]$$

with $u \sim \text{Uniform}[0,1]$. Let us explore the regimes of macroscopic parameters a, b, σ when either or both of these loss terms are low.

The impact of all the non-diagonal terms has been summed up in the "noise" ν. Though it is a deterministic function of $\mathbf{x}$, output i has no way to remove ν from $u + \nu$. The best it can do is to estimate u from $u + \nu$ because it doesn't have any other information. This exemplifies a key principle – by restricting the computational capacity of our decoder, deterministic, but complicated correlations act like noise.

The main advantage of the nonlinear autoencoder is that the dominant contribution to the loss, L_{off}, can be immediately screened away by making $a\nu + b$ either small or negative, allowing the network to focus on encoding active signals. This immediate screening is always possible either by choosing b large and negative or a and b small. However, these strategies come at a cost because the output value is distorted from u to $au + b$. It is thus preferable instead for ν and b to be as small as possible, which occurs when σ is as small as possible. As we will see, σ is the only parameter we are not free to choose in the large n_s limit, and whose value will be bounded as a function of n_d/n_s and p. Since L_{off} is the dominant contribution to the loss, therefore, it will thus be necessary to damp the signal by setting a small and/or b large and negative in regimes where σ is uncontrollably large.

Given that we see a statistical permutation symmetry in trained models let's consider symmetric strategies so that $W_{ii} = a$ and $\mathbf{b}_i = b$ for all $i = 1, \ldots, n_s$. We will show that optimizing the remaining macroscopic parameters makes $f_{\text{nonlinear}}$ act close to the identity on sparse inputs.

4.2 Optimizing the macroscopic parameters

We have seen qualitatively that a statistically symmetric strategy exists in certain regimes of the macroscopic parameters. Two of these parameters, a and b are unconstrained. Furthermore the loss should be monotonically increasing with σ because a larger σ implies more noise which hinders reconstruction. Thus we now prove lower bounds on σ and construct an artificial set of statistically permutation symmetric weights which achieve this bound. Finally we will compare the reconstruction loss of this strategy with the learned one to justify that those ignored microscopic degrees of freedom were indeed irrelevant.

4.2.1 Optimal σ

Assuming the permutation symmetry we discovered earlier in our empirical investigations we will derive a bound on the variance of the output. Additionally for an optimal choice of $\mathbf{b}$ the average loss is increasing in the variance, because a larger variance corresponds to a smaller signal-to-noise ratio. Taken together these two facts will give a lower bound on the loss. We will then provide an explicit construction which achieves this lower bound and illustrates how the algorithm works.

The lower bound on the variance comes from the fact that W is low-rank with constant diagonals. For now let us ignore the overall scale of W, and just rescale so that the diagonals are exactly 1. The bound we are about to prove is very similar to the Welch bound (Welch, 1974) who phrased it instead as a bound on the correlations between sets of vectors. We produce an argument for our context, which deals with potentially non-symmetric matrices W, the details of which are located in appendix D.

We show that

$$\sigma^2 \geq \frac{4p - 3p^2}{12}\left(\frac{n_s}{n_d} - 1\right) \tag{6}$$

with equality only when W is symmetric, maximum rank, with all non-zero eigenvalues equal. This naturally leads to a candidate for the optimal choice of W, namely matrices of the form

$$W \propto OPO^T \text{ and } W_{ii} = 1 \tag{7}$$

where O is an orthogonal matrix and P is any rank-n_d projection matrix. This kind of matrix saturates the bound because it is symmetric and has all nonzero eigenvalues equal to 1.

A note on the connections between these matrices and tight frames – if we take the "square root" of W as a $n_s \times n_d$ matrix $\sqrt{W}$ such that $\sqrt{W}\sqrt{W}^\dagger = W$ then the n_d dimensional rows of $\sqrt{W}$ are a tight-frame. This is because $\text{Tr}(W)^2 = n_s^2 = n_d \,\text{Tr}\, WW^\dagger$ which is the variational characterization of tight-frames as in Theorem 6.1 from Waldron (2018).

4.2.2 PERSIAN RUG MODEL

We now give an explicit construction of an optimal choice for W. The construction is based on a Hadamard matrix of size n_s. A Hadamard matrix H is one such that $H_{ij} = \pm 1$ and such that the rows of H are orthogonal.

We then define a Persian rug matrix as

$$R_{ij} = n_d^{-1} \sum_{k \in S} H_{ik} H_{jk}$$

where $S \subset \{1, ..., n_s\}$ with $|S| = n_d$. In fig. 1 we plot such a matrix for $n_s = 256$, $n_d = 40$, and S chosen randomly. The matrix R has diagonals equal to 1 because each diagonal is the average of n_d terms of the form $(H_{ik})^2 = (\pm 1)^2 = 1$. R is a projector because the rows of H are orthogonal, and therefore R is the sum of commuting rank-1 projectors. Therefore R saturates eq. (6). Furthermore, one can readily check that it exactly satisfies the symmetry for off-diagonal terms as well as shown in appendix F, which we direct readers to for a further discussion of R. There remain two variables to optimize, a and b (recall $\mathbb{E}[\nu]$ can be absorbed into b). We do this numerically and compare to a trained model in fig. 2 (details on the training process can be found in appendix G).

4.3 LOSS SCALING AT HIGH SPARSITY

Having obtained a simple expression for the loss in terms of constants a, b and two simple random variables $x \sim \text{Uniform}[0, 1]$ and $\nu \sim \mathcal{N}(0, \sigma)$, as well has having deduced an achievable lower bound for σ, we are now able to explain why the simple ReLU model performs so well at high sparsity. For ease of notation let us use $r = n_d/n_s$.

4.3.1 INITIAL LOSS (RATIO=0)

Let us first consider the $r \to 0$ limit with all other parameters fixed. Then $\sigma \to \infty$ because of the bound in eq. (6) so the fluctuations in ν overwhelms the signal term. This means that the optimal a is

$$a = p \frac{\mathbb{E}_{u,\nu}\left[u \,\text{ReLU}(\nu + b)\right]}{\mathbb{E}_{\nu}\left[\text{ReLU}(\nu + b)^2\right]} + O(\sigma^{-1}). \tag{8}$$

The loss then becomes

$$\begin{aligned} L &= (1-p)a^2 \mathbb{E}_{\nu}[\text{ReLU}\,(\nu + b)^2] + p\mathbb{E}_{u,\nu}[(u - a\,\text{ReLU}\,(\nu + b))^2] + O(\sigma^{-1}) \\ &= a^2 \mathbb{E}_{\nu}[\text{ReLU}\,(\nu + b)^2] - 2ap\mathbb{E}_{u,\nu}[u\,\text{ReLU}(\nu + b)] + p\mathbb{E}_u[u^2] + O(\sigma^{-1}) \end{aligned}$$

plugging in a explicitly gives

$$L = p\mathbb{E}_u[u^2] - p^2 \frac{\left(\mathbb{E}_u\left[u\right]\right)^2 \left(\mathbb{E}_{\nu}\left[\text{ReLU}(\nu + b)\right]\right)^2}{\mathbb{E}_{\nu}\left[\text{ReLU}(\nu + b)^2\right]} + O(\sigma^{-1}).$$

Thus we can conclude that

$$\lim_{r \to 0} L = p\mathbb{E}_u[u^2] + O(p^2) = \frac{p}{3} + O(p^2)$$

Thus we see that in the $p \ll 1$ regime we have $L \to L_0(p) \sim O(p)$ independent of the other parameters. We will now see that increasing r will quickly cause the loss to drop to $O(p^2)$.

4.3.2 DERIVING THE LOSS SCALING

In appendix E.1 we derive an upper bound on the loss function by plugging in appropriate ansatze for a and b. We find that

$$L < O\left(\sigma^2 p \log \frac{1}{p}\right) \sim O\left(\frac{p^2}{r} \log \frac{1}{p}\right), \tag{9}$$

when p is small and when $r \gg p$.

On the other hand, in appendix E.2 we also derive a lower bound in the high-sparsity limit $L >$

$O(p^2/r)$ in the high sparsity limit up to logarithmic corrections. We show this in fact holds for a more general class of activation functions. In particular, any function which acts element-wise or filters out elements will give an on-loss contribution of the form

$$\mathbb{E}_{u,\nu}[(u - f(u + \nu + b))^2]$$

which has a lower bound due to ν destroying information about u. Thus we can conclude that

$$L \sim O\left(\frac{p^2}{r}\right)$$

up to logarithmic factors whenever $\frac{p}{r} \ll 1$. It is sensible that the loss function scales inversely with compression ratio.

5 Discussion and Relationship to other works

5.1 Autoencoders

Our work focuses specifically on sparse autoencoders, and encoding sparse data, which is parallel to work explaining the dynamics and emergence of feature learning in autoencoders. Refinetti & Goldt (2023) show that shallow autoencoders learn the principal components of the data sequentially and Nguyen (2021) shows a similar dynamical result via a mean-field analysis. It's seen that such autoencoders function even in a regime where the number of features and the size of the input are proportional with numerical evidence for Gaussian universality (Shevchenko et al., 2023). This universality is shown for shallow in auto-encoders following gradient dynamics (Kögler et al., 2024).

5.1.1 Mechanistic Interpretability and Sparse Autoencoders

Mechanistic interpretability is a research agenda which aims to understand learned model algorithms through studying their weights (see Olah et al. (2020) for an introduction). Recent results relating to language models include Meng et al. (2023), which finds a correspondence between specific facts and feature weights, along with Olsson et al. (2022), which shows that transformers learn in context learning through the mechanism of "induction heads".

A key issue for the agenda of mechanistic interpretability is that the model stores features in superposition. Elhage et al. (2022) introduced the toy model of superposition we study in this paper. While that work focused on mapping empirically behaviors of the model in multiple regimes of interest such as correlated inputs, we focused on a regime with enough symmetry to solve the model analytically given observed symmetries in trained models. Chen et al. (2023) study this model in the context of singular learning theory. As part of their work, they characterize the loss using a different high sparsity approximation than the one we present in this paper (they assume exactly one input feature is on). Then they derive a subset of the critical points and their corresponding local learning coefficients under the assumption $n_d = 2$. Refinetti & Goldt (2023) study the learning dynamics of the same model but without the sparsity assumption.

One way to extract interpretable features that are stored in superposition is through dictionary learning. While the concept of dictionary learning was introduced by (Mallat & Zhang, 1993), the practical use of sparse autoencoders to understand large language models has accelerated recently due to mezzo-scale open weight models (Gao et al., 2024; Lieberum et al., 2024) and large-scale open-output models Bricken et al. (2023). These features are highly interpretable (Cunningham et al., 2023) and scale predictably. Interestingly, the scaling is quite similar for the various different architectures they consider, differing primarily by a constant, which fits with the predictions in this work.

Our study of Elhage et al. (2022)'s model of superposition lend some insight into the dictionary learning problem. In particular, we have seen that the dominant source of error is not from determining which features are present, but rather the actual values of those features. Small modifications to the activation functions, such as gating Rajamanoharan et al. (2024), k-sparse Makhzani & Frey (2013), or TRec non-linearity Taggart (2024); Konda et al. (2014), are insufficient to fix this problem as they do not solve the basic issue of noisy outputs. In this context our work implies that innovative

architectures, that are suitable both for gradient-based training and also for decoding sparse features, must be developed.

While our work focuses on Elhage et al. (2022) toy model of superposition, one can study the dictionary learning with sparse autoencoders problem directly under various models for the data and various algorithms (see, for example, Rangamani et al. (2017); Nguyen et al. (2019); Arora et al. (2015); Agarwal et al. (2016); Spielman et al. (2012)). For these problems, we suppose we are given data vectors y_i generated by $y_i = A^* x_i$, where $A^* \in \mathbb{R}^{n_d \times n_s}$ is the "true" dictionary and x_i are parametrically sparse vectors. The goal is then to recover A^* and the x_i. Under various assumptions on A^* and the x_i, one can prove various desirable results for various algorithms for estimating them. For example, in Rangamani et al. (2017) and Nguyen et al. (2019) it is assumed that A^* has unit columns and is *incoherent*, meaning that its columns $\{A_i^*\}$ have inner-products bounded by

$$\max_{i \neq j} |\langle A_i^*, A_j^* \rangle| \leq \frac{\mu}{\sqrt{n}}.$$

These authors then give convergence results for learning the model

$$\hat{y} = V^T ReLU(Vy - \epsilon) \tag{10}$$

with ϵ a learnable bias and $V \in \mathbb{R}^{n_s \times n_d}$ learnable weights. In particular, Rangamani et al. (2017) shows that the support of x can be recovered for sufficiently sparsity and incoherence and that A^* is critical point for V in the loss landscape; Nguyen et al. (2019) shows that eq. (10) trained with gradient descent recovers the true dictionary in certain parameter regimes.

In contrast, the neural networks in our work search over the space of dictionaries to find ones that encode sparse information in a way particularly suitable for reconstruction by a single linear + ReLU layer. As a result, the dictionary our network finds contains *additional structure optimized for a particular recovery process*. For this reason, we find that the relevant error parameter is not the incoherence (in our notation) $\max_{i \neq j} |W_{ij}|$ but rather the variance of off-diagonal elements in each row $\sum_j W_{ij}^2$, and that it is this parameter that needs to be minimized for a given compression ratio.

5.1.2 Compressed Sensing, Statistical Physics

It is known that compressed sparse data can be exactly reconstructed by solving a convex problem (Candes & Tao, 2005; Candes et al., 2006; Donoho & Elad, 2003; Donoho, 2006) given knowledge of the compression matrix. Furthermore, using tools from statistical physics it is possible to show that this holds for typical compressed sparse data (Ganguli & Sompolinsky, 2010). Learning the compression matrix is also easy in certain circumstances(Sakata & Kabashima, 2013). For a more general review on compressed sensing and it's history consider the introduction by Davenport et al. (2012). The reconstruction procedure typically used in compressed sensing is optimizing a (convex) relaxation of finding the sparsest set of features which reproduces your data vector. This is significantly different to the setting of sparse autoencoders which try to obtain the sparse features using only one linear + activation layer.

The discrepancy between the ability of convex optimization techniques to achieve zero loss while a linear + ReLU model necessarily incurs an error suggests that a more complex model architecture is needed for sparse autoencoders when it is desirable to calculate the feature magnitude to high precision. This may occur, for example, if one wishes to insert a sparse autoencoder into a model without corrupting its downstream outputs. An important line of work is algorithms based on message-passing schemes brought to fame by Donoho et al. (2009), and extended to more general encoding matrices by Rangan et al. (2019), a more general encoding scheme (Schniter et al., 2016), for ill conditioned matrices (Ma & Ping, 2017), and proved without statistical physics methods by Takeuchi (2019). These works may hold the key to improving interpretability, particularly for downstream tasks such as circuit recovery.

REFERENCES

Alekh Agarwal, Animashree Anandkumar, Prateek Jain, and Praneeth Netrapalli. Learning sparsely used overcomplete dictionaries via alternating minimization. SIAM Journal on Optimization, 26 (4):2775–2799, 2016.

Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. Simple, efficient, and neural algorithms for sparse coding. In Conference on learning theory, pp. 113–149. PMLR, 2015.

Christopher M Bishop and Nasser M Nasrabadi. Pattern recognition and machine learning, volume 4. Springer, 2006.

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.

E.J. Candes and T. Tao. Decoding by linear programming. IEEE Transactions on Information Theory, 51(12):4203–4215, 2005. doi: 10.1109/TIT.2005.858979.

E.J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on Information Theory, 52 (2):489–509, 2006. doi: 10.1109/TIT.2005.862083.

Zhongtian Chen, Edmund Lau, Jake Mendel, Susan Wei, and Daniel Murfet. Dynamical versus bayesian phase transitions in a toy model of superposition, 2023. URL https://arxiv.org/abs/2310.06301.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. arXiv preprint arXiv:2309.08600, 2023.

Mark A Davenport, Marco F Duarte, Yonina C Eldar, and Gitta Kutyniok. Introduction to compressed sensing., 2012.

David L Donoho. Compressed sensing. IEEE Transactions on information theory, 52(4):1289–1306, 2006.

David L. Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ¡sup¿1¡/sup¿ minimization. Proceedings of the National Academy of Sciences, 100(5):2197–2202, 2003. doi: 10.1073/pnas.0437847100. URL https://www.pnas.org/doi/abs/10.1073/pnas.0437847100.

David L Donoho, Arian Maleki, and Andrea Montanari. Message-passing algorithms for compressed sensing. Proceedings of the National Academy of Sciences, 106(45):18914–18919, 2009.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. Transformer Circuits Thread, 2022. URL https://transformer-circuits.pub/2022/toy_model/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.

Surya Ganguli and Haim Sompolinsky. Statistical mechanics of compressed sensing. Physical review letters, 104(18):188701, 2010.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. arXiv preprint arXiv:2406.04093, 2024.

Kevin Kögler, Alexander Shevchenko, Hamed Hassani, and Marco Mondelli. Compression of structured data with autoencoders: Provable benefit of nonlinearities and depth. arXiv preprint arXiv:2402.05013, 2024.

Kishore Konda, Roland Memisevic, and David Krueger. Zero-bias autoencoders and the benefits of co-adapting features. arXiv preprint arXiv:1402.3337, 2014.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2, 2024. URL https://arxiv.org/abs/2408.05147.

Junjie Ma and Li Ping. Orthogonal amp. IEEE Access, 5:2020–2033, 2017.

Alireza Makhzani and Brendan Frey. K-sparse autoencoders. arXiv preprint arXiv:1312.5663, 2013.

Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. IEEE Transactions on signal processing, 41(12):3397–3415, 1993.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models, 2024. URL https://arxiv.org/abs/2403.19647.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt, 2023. URL https://arxiv.org/abs/2202.05262.

Phan-Minh Nguyen. Analysis of feature learning in weight-tied autoencoders via the mean field lens. arXiv preprint arXiv:2102.08373, 2021.

Thanh V. Nguyen, Raymond K. W. Wong, and Chinmay Hegde. On the dynamics of gradient descent for autoencoders. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, volume 89 of Proceedings of Machine Learning Research, pp. 2858–2867. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/nguyen19a.html.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. Distill, 2020. doi: 10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads, 2022. URL https://arxiv.org/abs/2209.11895.

VV Petrov. of independent random variables. Yu. V. Prokhorov. V. StatuleviCius (Eds.), 1972.

Elad Plaut. From principal subspaces to principal components with linear autoencoders, 2018. URL https://arxiv.org/abs/1804.10253.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Improving dictionary learning with gated sparse autoencoders. arXiv preprint arXiv:2404.16014, 2024.

Akshay Rangamani, Anirbit Mukherjee, Amitabh Basu, Tejaswini Ganapathy, Ashish Arora, Sang Chin, and Trac D. Tran. Sparse coding and autoencoders, 2017. URL https://arxiv.org/abs/1708.03735.

Sundeep Rangan, Philip Schniter, and Alyson K Fletcher. Vector approximate message passing. IEEE Transactions on Information Theory, 65(10):6664–6684, 2019.

Maria Refinetti and Sebastian Goldt. The dynamics of representation learning in shallow, non-linear autoencoders. Journal of Statistical Mechanics: Theory and Experiment, 2023(11):114010, 2023.

Ayaka Sakata and Yoshiyuki Kabashima. Statistical mechanics of dictionary learning. Europhysics Letters, 103(2):28008, 2013.

Philip Schniter, Sundeep Rangan, and Alyson K Fletcher. Vector approximate message passing for the generalized linear model. In 2016 50th Asilomar conference on signals, systems and computers, pp. 1525–1529. IEEE, 2016.

Aleksandr Shevchenko, Kevin Kögler, Hamed Hassani, and Marco Mondelli. Fundamental limits of two-layer autoencoders, and achieving them with gradient methods. In International Conference on Machine Learning, pp. 31151–31209. PMLR, 2023.

Daniel A Spielman, Huan Wang, and John Wright. Exact recovery of sparsely-used dictionaries. In Conference on Learning Theory, pp. 37–1. JMLR Workshop and Conference Proceedings, 2012.

Glen M. Taggart. Prolu: A nonlinearity for sparse autoencoders. `https://www.alignmentforum.org/posts/HEpufTdakGTTKgoYF/prolu-a-nonlinearity-for-sparse-autoencoders`, 2024.

Keigo Takeuchi. Rigorous dynamics of expectation-propagation-based signal recovery from unitarily invariant measurements. IEEE Transactions on Information Theory, 66(1):368–386, 2019.

Shayne FD Waldron. An introduction to finite tight frames. Springer, 2018.

Lloyd Welch. Lower bounds on the maximum cross correlation of signals (corresp.). IEEE Transactions on Information theory, 20(3):397–399, 1974.

A Convergence Figures of Section 3.2

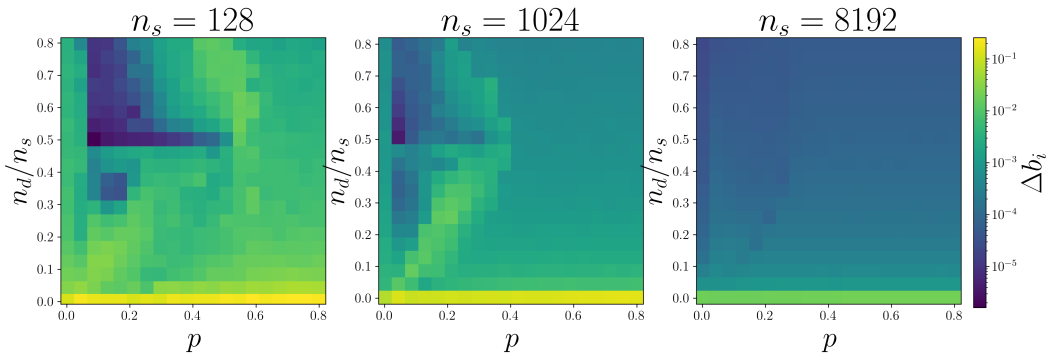

Figure 5: Permutation symmetry of bias values. We plot the mean-square fluctuation of values in the bias vectors corresponding to each model, which are trained as a function of p and n_d/n_s. As n_s increases the fluctuation over bias elements generally decreases in all trained models.

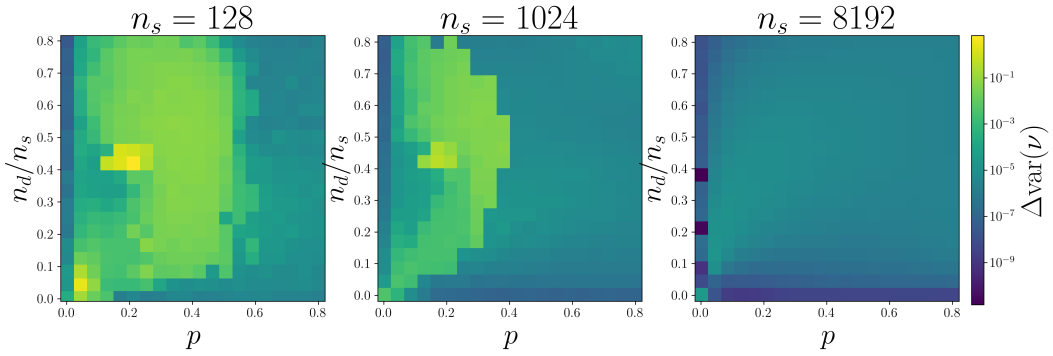

Figure 6: Permutation symmetry of off-diagonal statistics. The symmetry breaking parameter $\Delta\,\mathrm{var}(\nu)$ is given by the variance across all rows of the squared sum of the off diagonal elements in each row, up to a constant. Once n_s reaches 8192 all noise variables have nearly identical variances.

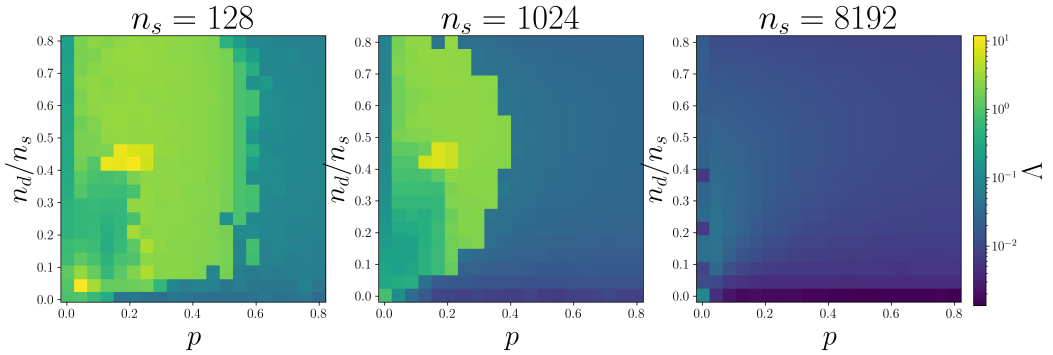

Figure 7: Interference Gaussianity in the large n_s limit. We show that the Lyapunov condition is better satisfied as n_s grows larger, by plotting the worst-case Lyapunov condition (see eq. (3)) over all rows of W. Models are trained at different p and n_d/n_s values. We see that for small n_s only models trained at large p have nearly-uniform off-diagonal entries whereas all models approach uniformity at large n_s.

B OPTIMIZING OVER a

We may optimize over a analytically because it appears almost quadratically in the loss. Consider the expression for the average loss from eq. (5) with b replaced with $a \cdot b$. As long as $a \neq 0$ this redefinition doesn't change the set of accessible models.

Furtheremore let us restrict to positive a which allows us to rewrite the loss as

$$L = \mathbb{E}_{x,\nu}[\big(x - \mathrm{ReLU}\,(a(x+\nu+b))\big)^2] = \mathbb{E}_{x,\nu}[\big(x - a\,\mathrm{ReLU}\,(x+\nu+b)\big)^2]. \tag{11}$$

The restriction to positive a is acceptable because we never see negative off-diagonal elements in our trained models. Now, optimizing over a is exactly linear regression; we can obtain the optimal value of a with the standard method

$$0 = \frac{d}{da}L = -2\mathbb{E}\left[(x - a\,\mathrm{ReLU}(x+\nu+b))\,\mathrm{ReLU}(x+\nu+b)\right] \tag{12}$$

which implies that

$$a_{\text{opt}} = \frac{\mathbb{E}_{x,\nu}\left[x\,\mathrm{ReLU}(x+\nu+b)\right]}{\mathbb{E}_{x,\nu}\left[\mathrm{ReLU}(x+\nu+b)^2\right]}. \tag{13}$$

Notice that the optimal a is always positive, which is consistent with the assumption we made earlier.

C BOUNDING THE RECONSTRUCTION ERROR

On term:

We can start to write the on term similarly as

$$\langle (u - \mathrm{ReLU}\,(u+\nu))^2 \rangle = \left\langle \sigma^2 \int_{-\infty}^{\infty} d\nu \frac{e^{-\frac{(\nu+\frac{|b|}{\sigma})^2}{2}}}{\sqrt{2\pi}} (u - \mathrm{ReLU}(u+\sigma\nu))^2 \right\rangle_u .$$

Now we write the integral in two parts to get rid of the ReLU: one when $u + \sigma\nu < 0$ and one when $u + \sigma\nu > 0$. This gives

$$\underbrace{\left\langle \int_{-\infty}^{-\frac{u}{\sigma}} d\nu \frac{e^{-\frac{(\nu+\frac{|b|}{\sigma})^2}{2}}}{\sqrt{2\pi}} u^2 \right\rangle_u}_{E_r} + \underbrace{\left\langle \sigma^2 \int_{-\frac{u}{\sigma}}^{\infty} d\nu \frac{e^{-\frac{(\nu+\frac{|b|}{\sigma})^2}{2}}}{\sqrt{2\pi}} \nu^2 \right\rangle_u}_{E_\nu} .$$

Where the first term E_r represents error coming from the ReLU and the second term E_ν represents error coming from the noise. The scaling of E_ν can be easily bounded:

$$E_\nu < \sigma^2 \int_{-\infty}^{\infty} d\nu \frac{e^{-\frac{(\nu+\frac{|b|}{\sigma})^2}{2}}}{\sqrt{2\pi}} \nu^2 \sim O(\sigma^2 + \sigma b + b^2)$$

And thus we see, unsurprisingly, that we need to set $b \ll 1$ to get a good bound.

To upper bound E_r write the u integral in two intervals: $[0, 2|b|]$ and $[2|b|, 1]$, corresponding to regions in which the interval of the ν integral does and "decisively" does not include the the mean respectively. In particular, we have

$$E_r < \underbrace{\int_0^{2|b|} u^2 du \int_{-\infty}^{-\frac{u}{\sigma}} d\nu \frac{e^{-\frac{(\nu+\frac{|b|}{\sigma})^2}{2}}}{\sqrt{2\pi}}}_{E_r^{\text{mean}}} + \underbrace{\int_{2|b|}^{1} u^2 du \int_{-\infty}^{-\frac{u}{\sigma}} d\nu \frac{e^{-\frac{(\nu+\frac{|b|}{\sigma})^2}{2}}}{\sqrt{2\pi}}}_{E_r^{\text{tail}}} .$$

Since in E_r^{mean} the ν integrals' interval includes the mean we may as well extend the interval to the full real line to get the bound, giving

$$E_r^{\text{mean}} < \int_0^{2|b|} u^2 du = O(b^3).$$

E_r^{tail} can be upper bounded by setting the ν range to the maximum value of $2|b|$, so we have

$$E_r^{\text{tail}} < (1-2|b|)\int_{-\infty}^{-\frac{2|b|}{\sigma}} d\nu \frac{e^{-\frac{(\nu+\frac{|b|}{\sigma})^2}{2}}}{\sqrt{2\pi}} = (1-2|b|)\int_{-\infty}^{0} d\nu \frac{e^{-\frac{(\nu-\frac{|b|}{\sigma})^2}{2}}}{\sqrt{2\pi}} < O(1)(1-2|b|)e^{-\frac{|b|^2}{2\sigma^2}}.$$

Putting it all together gives

$$L < (1-p)O(\sigma^2 e^{-\frac{b^2}{2\sigma^2}}) + pO(b^3 + e^{-\frac{|b|^2}{2\sigma^2}} + be^{-\frac{|b|^2}{2\sigma^2}} + \sigma^2 + \sigma b + b^2).$$

Plugging in the b scaling from eq. (23) and keeping only the lowest order terms gives

$$L < O(\sigma^2 p \log\frac{1}{p}) \sim O(\frac{p^2}{r}\log\frac{1}{p}). \tag{14}$$

D Minimal Variance Bound

We will show a minimum variance bound for matrices W which have all diagonals equal to 1 and also have maximum rank n_d. In this case we know that $\operatorname{Tr} W = n_s$. On the other hand we also know that the trace is the sum of the eigenvalues, and because W has rank at most n_d that

$$n_s = \sum_{i=1}^{n_d} \lambda_i \tag{15}$$

for the eigenvalues λ_i of W. Now we solve for the mean of the variance across rows,

$$\frac{1}{n_s}\sum_{i=1}^{n_s}\operatorname{Var}(\nu_i) = \frac{4p-3p^2}{12n_s}\sum_{i=1}^{n_s}\sum_{j=1,j\neq i}^{n_s} W_{ij}^2 = \frac{4p-3p^2}{12n_s}\left(\operatorname{Tr}(WW^\dagger) - n_s\right). \tag{16}$$

Here the first equality arises from the definition of ν_i (remembering that we have set the diagonals to 1 exactly) and substituting the variance of x_j, while the second equality follows because $\operatorname{Tr}(WW^\dagger)$ is the sum of the square of all entries of W, and we subtract off the diagonal entries.

Because we want a bound on this quantity related to the eigenvalues of W, it is convenient to use the Schur decomposition of $W = QUQ^\dagger$. Here Q is a unitary matrix and U is upper-triangular with the eigenvalues of W on the diagonal. This allows us to lower bound the trace

$$\operatorname{Tr}(WW^\dagger) = \operatorname{Tr}(QUQ^\dagger QU^\dagger Q^\dagger) = \operatorname{Tr}(UU^\dagger) = \sum_{i,j=1}^{n_s}|U_{ij}|^2 \geq \sum_{i=1}^{n_d}|\lambda_i|^2 \geq \frac{n_s^2}{n_d} \tag{17}$$

where the last inequality follows from Cauchy-Schwarz and eq. (15). With this we find a bound on the variance

$$\frac{1}{n_s}\sum_{i=1}^{n_s}\operatorname{Var}(\nu_i) \geq \frac{4p-3p^2}{12}\left(\frac{n_s}{n_d}-1\right), \tag{18}$$

with equality if W is symmetric with all non-zero eigenvalues equal. These two conditions follow because the two inequalities in the proof become equalities when these conditions are met. This naturally leads to a candidate for the optimal choice of W, namely matrices of the form

$$W \propto OPO^T \text{ and } W_{ii} = 1 \tag{19}$$

where O is an orthogonal matrix and P is any rank-n_d projection matrix. This kind of matrix saturates both bounds because it is symmetric and has all nonzero eigenvalues equal to 1.

E Bounds on the Loss

E.1 Upper bound on loss scaling

We now show that the loss drops off quickly in the sense that for $\frac{r}{p} \gg 1$ we get that $L(p)/p \to 0$, i.e. $L(p)$ scales super-linearly with p. We will consider the regime where $r \ll 1$ holds[2] so that we

[2]For example $r = p^{1-\epsilon}$ for any $\epsilon \in (0,1)$.

may take $\sigma^2 \sim \frac{p}{r} \ll 1$.

To obtain the upper bound we will make educated estimates for values of a and b that are near optimal. In particular, in appendix B we show that the optimal value of a is (after absorbing the mean of ν into b):

$$a_{\text{opt}} = \frac{\mathbb{E}_{x,\nu}\left[x\,\text{ReLU}(x+\nu)\right]}{\mathbb{E}_{x,\nu}\left[\text{ReLU}(x+\nu)^2\right]}. \tag{20}$$

From the form of the loss, we know that b must decrease as p decreases for the loss to go down faster than $O(p)$. Thus ν has both a mean and variance approaching 0, and $a_{\text{opt}} \to 1$. Thus we plug in $a = 1$ before taking these limits in the expectation of getting a good upper bound. The loss then takes the form

$$L = (1-p)L_{\text{off}} + pL_{\text{on}}$$

with

$$L_{\text{off}} = \mathbb{E}\left[\text{ReLU}\left(\nu\right)^2\right], \text{ and}$$
$$L_{\text{on}} = \mathbb{E}\left[\left(u - \text{ReLU}\left(u+\nu\right)\right)^2\right]$$

Off term: The off term can be upper bounded via

$$\mathbb{E}\left[\text{ReLU}(\nu)^2\right] = \int_0^\infty d\nu \frac{e^{-\frac{(\nu+|b|)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\nu^2 = \sigma^2 \int_0^\infty d\nu \frac{e^{-\frac{(\nu+\frac{|b|}{\sigma})^2}{2}}}{\sqrt{2\pi}}\nu^2 \tag{21}$$
$$\leq \frac{\sigma^2}{2} e^{-\frac{b^2}{2\sigma^2}} \tag{22}$$

and thus we see we need to set $\frac{b}{\sigma} \gg 1$ to get a good bound. In particular, we know empirically that the loss drop happens at increasingly smaller r. To ensure this we let $\sigma^2 \sim \frac{p}{r}$ scale at some rate slower than p. Thus to ensure that the total loss decreases faster than $O(p)$, we need $e^{-\frac{b^2}{2\sigma^2}} \sim O(p)$ or in other words

$$b \sim \sigma\sqrt{\log\frac{1}{p}}. \tag{23}$$

On term: We perform a similar, but slightly more involved computation in appendix C and combine with the off term to obtain

$$L < (1-p)O(\sigma^2 e^{-\frac{b^2}{2\sigma^2}}) + pO(b^3 + e^{-\frac{|b|^2}{2\sigma^2}} + be^{-\frac{|b|^2}{2\sigma^2}} + \sigma^2 + \sigma b + b^2).$$

Plugging in the b scaling from eq. (23) and keeping only the lower order terms gives

$$L < O\left(\sigma^2 p \log\frac{1}{p}\right) \sim O\left(\frac{p^2}{r}\log\frac{1}{p}\right). \tag{24}$$

E.2 Lower bound on loss scaling

We now show a lower bound on the loss in the $p \to 0$ limit. To do this, we will show a more general lower bound on the on loss for any deterministic function of the pre-activation. Specifically, we would like to lower bound

$$L \leq pL_{\text{on}} = \mathbb{E}[(u - f\left(u+\nu\right))^2]$$

for any function f with $u \sim \text{Uniform}[0,1]$ and $\nu \sim \mathcal{N}(0,\sigma)$. Recall that there is no need to consider the bias b as it can be absorbed into ν. Recall that the optimal function f is given by

$$f^*(u+\nu) = \mathbb{E}[u|u+\nu].$$

Let's make a change of variable from u, ν to $y \equiv u+\nu, u$, and then use the tower property to rewrite L_{on} as

$$L_{\text{on}} = \mathbb{E}_{y\sim P_{u+\nu}}\left[\mathbb{E}_{u|y}\left[(u - f^*(u+\nu))^2\right]\right]. \tag{25}$$

We first draw y from the marginal distribution of $u+\nu$ and then draw u from the conditional distribution given y. Because f^* is exactly the conditional expectation the interior expectation becomes the conditional variance

$$L_{\text{on}} = \mathbb{E}_y\left[\text{Var}\left[u|y\right]\right]. \tag{26}$$

Because we want to lower bound L_{on} it will be convenient to start with a lower bound for the conditional variance. We will lower bound the conditional variance for $y \in [\sigma, 1-\sigma]$, and then use that lower bound to find a lower bound for the loss, with a goal of showing that the loss is lower bounded by a constant multiple of σ^2, for $\sigma < \frac{1}{4}$. This will show that the overall loss of any strategy, even one which can perfectly estimate which features are on or off, is incapable of achieving a reconstruction error better than $O(p^2/r)$.

The conditional distribution for u is a truncated Gaussian distribution. By Bayes' theorem

$$P[u|u+\nu=y] = \frac{P[u+\nu=y]P[u]}{P[y]} \tag{27}$$

$$= \begin{cases} \frac{e^{-(u-y)^2/2\sigma^2}}{\int_0^1 dx e^{-(x-y)^2/2\sigma^2}} & \text{if } u \in [0,1] \\ 0 & \text{otherwise,} \end{cases} \tag{28}$$

with normalizing constant $Z(y) = \int_0^1 dx e^{-(x-y)^2/2\sigma^2} < \sqrt{2\pi\sigma^2}$. This is a truncated Gaussian distribution. Fix $y \in [\sigma, 1-\sigma]$ so that all distributions are implicitly conditioned on y for now. Sample u via the following procedure. First we decide if $|u-y| \leq \sigma$ and then we either sample from the conditional distribution $P[u|y$ and $|u-y| \leq \sigma]$ or $P[u|y$ and $|u-y| \geq \sigma]$ with their corresponding probabilities. Let R be the indicator random variable denoting $|u-y| \leq \sigma$. Then by the law of total variance

$$\text{Var}\left[u \mid y\right] = P[R=1]\,\text{Var}\left[u \mid R=1\right] + P[R=0]\,\text{Var}\left[u \mid R=0\right] + \text{Var}_R\left[\mathbb{E}[u \mid R]\right] \tag{29}$$

$$\geq P[R=1]\,\text{Var}\left[u \mid R=1\right] \tag{30}$$

where we have dropped the latter two positive terms to derive the lower bound. $P[R=1] \geq \text{erf}(2^{-1/2})$ because the chance a truncated Gaussian is within one σ of its mode is larger than that for an untruncated Gaussian, given that the truncation is more than σ away from the mode. This condition is satisfied by construction because we have chosen y to be more than σ from the boundary.

Additionally a trivial scaling argument shows that the variance is proportional to σ^2 which means that there is some constant, $C > 0$ such that

$$\text{Var}\left[u \mid y\right] \geq C\sigma^2 \tag{31}$$

when $y \in [\sigma, 1-\sigma]$. To complete the argument we now return to

$$L_{\text{on}} = \mathbb{E}_y\left[\text{Var}\left[u|y\right]\right] \geq \mathbb{E}_y\left[\text{Var}\left[u|y\right] 1_{y\in[\sigma,1-\sigma]}\right] \geq C\sigma^2 P[y \in [\sigma, 1-\sigma]]. \tag{32}$$

For $\sigma = 1/4$ this probability is clearly finite and for $\sigma < 1/4$ it is increasing as σ decreases so it is uniformly bounded below by a constant C_1. So finally

$$L_{\text{on}} \geq C'\sigma^2 \implies L \geq C'p\sigma^2 \approx \frac{C'p^2}{r}. \tag{33}$$

F Persian Rug Construction Satisfies Permutation Symmetry

In this section we provide a short discussion on Hadamard matrices, and more importantly proofs that our Persian Rug construction satisfies the permutation symmetry conditions.

A $n \times n$ matrix H is a Hadamard matrix if every entry of H is either 1 or -1, and if all the rows of

H are orthogonal. This implies that $HH^T = nI$ where I is the identity matrix.

As a reminder we construct the rug matrix of rank n_d and size n_s by first choosing a subset $S \subseteq \{1, \ldots, n_s\}$ of size $|S| = n_d$. Then we construct

$$R_{ij} = \frac{1}{n_d} \sum_{k \in S} H_{ik} H_{jk} \tag{34}$$

for any Hadamard matrix H of dimension n_s

These properties are sufficient for us to prove the required symmetries, as well as the spectral properties of R:

- $R_{ii} = 1$,
- For any $i = 1, \ldots n_s$ that $\sum_{j=1, j \neq i}^{n_s} R_{ij}^2 = \frac{n_s}{n_d} - 1$,
- R is proportional to a projector.

The first property is apparent from the fact that all entries of H are ± 1.

$$R_{ii} = \frac{1}{n_d} \sum_{k \in S} (H_{ik})^2 = \frac{1}{n_d} \sum_{k \in S} (\pm 1)^2 = 1. \tag{35}$$

The second property follows similarly, but with some more algebra. Without loss of generality let $i = 1$ so that we consider the first row's off-diagonal terms and let δ_{ij} denote the Kronecker delta symbol. Then their sum is

$$\sum_{j=2}^{n_s} R_{1j}^2 = \frac{1}{n_d^2} \sum_{j=2}^{n_s} \sum_{k_1, k_2 \in S} H_{1k_1} H_{1k_2} H_{jk_1} H_{jk_2} \tag{36}$$

$$= \frac{1}{n_d^2} \sum_{k_1, k_2 \in S} H_{1k_1} H_{1k_2} \sum_{j=2}^{n_s} H_{jk_1} H_{jk_2} \tag{37}$$

$$= \frac{1}{n_d^2} \sum_{k_1, k_2 \in S} H_{1k_1} H_{1k_2} (H_{\cdot k_1} \cdot H_{\cdot k_2} - H_{1k_1} H_{1k_2}) \tag{38}$$

where we use the notation $H_{\cdot k}$ for the kth row of the matrix H viewed as a vector. We know these rows are orthogonal and have norm n_s because all their entries are ± 1 so

$$\sum_{j=2}^{n_s} R_{1j}^2 = \frac{1}{n_d^2} \sum_{k_1, k_2 \in S} H_{1k_1} H_{1k_2} (n_s \delta_{k_1 k_2} - H_{1k_1} H_{1k_2}) \tag{39}$$

$$= \frac{n_s}{n_d^2} \sum_{k_1, k_2 \in S} H_{1k_1} H_{1k_2} \delta_{k_1 k_2} - \frac{1}{n_d^2} \sum_{k_1, k_2 \in S} H_{1k_1} H_{1k_2} H_{1k_1} H_{1k_2} \tag{40}$$

$$= \frac{n_s}{n_d^2} \sum_{k \in S} H_{1k}^2 - \frac{1}{n_d^2} \sum_{k_1, k_2 \in S} H_{1k_1}^2 H_{1k_2}^2 = \frac{n_s}{n_d} - 1, \tag{41}$$

where we used the fact $|S| = n_d$ and the unit norm of the entries to simplify. This shows that the noise ν_i has the same variance over all rows.

Finally we show that R is proportional to a projector. Looking at

$$(R^2)_{ij} = \frac{1}{n_d^2} \sum_{l=1}^{n_s} R_{il} R_{lj} \tag{42}$$

$$= \frac{1}{n_d^2} \sum_{k_1, k_2 \in S} \sum_{l=1}^{n_s} H_{ik_1} H_{lk_1} H_{lk_2} H_{jk_2} \tag{43}$$

$$= \frac{1}{n_d} \sum_{k_1, k_2 \in S} H_{ik_1} \delta_{k_1 k_2} H_{jk_2} \tag{44}$$

$$= \frac{1}{n_d} \sum_{k \in S} H_{ik} H_{jk} = R_{ij} \tag{45}$$

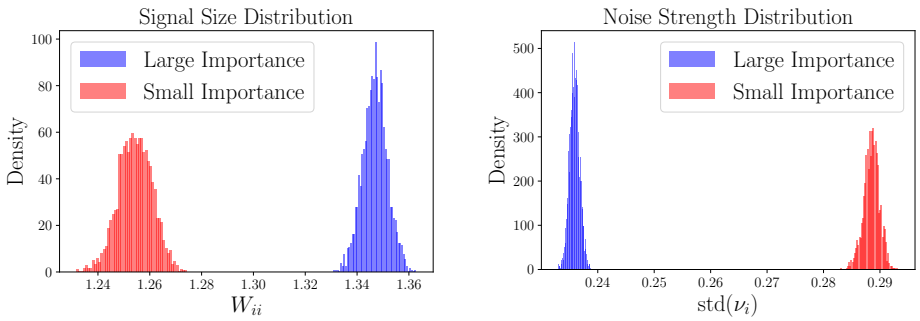

Figure 8: Distributions of diagonal terms (of W) in a single trained model with $n_s = 4096$ sparse features, $n_d = 1024$ dense dimensions, and relative importance weight $\alpha = 1/2$ for the less important features. The first subfigure shows that the distribution of diagonal components with small importance (red) and large importance (blue) are separated, but similar in magnitude. On the other hand The distribution of the noises is different, with more noise allocated to the less important features.

The fact $R^2 = R$ means that R is a projector, and hence it is proportional to a projector.

G Training Details

For all of the toy models (eq. (1) with $n_s \in \{128, 1024, 8192\}$) and use a learning rate of $3 \times 10^{-3}/\sqrt{n_s}$. For the Hadamard model ($n_s = 8192$), we use the same stopping strategy described for the toy models. We use a batch size of 512, maximum number of epochs of 100, a learning rate of $3 \times 10^{-1}/\sqrt{n_s}$, and also train 5 models and keep the model with the lowest loss.

H Partially Breaking Permutation Symmetry

A natural question following our analysis; To what extent do the qualitative features of the results we derive depend on the permutation symmetry of the input vectors $\mathbf{x}$? To move away from this assumption a little, we perform numerical experiments on the following loss

$$L = (\mathbf{x}; W_{\text{out}}, W_{\text{in}}, \mathbf{b}) = n_s^{-1} \sum_{i=1}^{n_s} M_i (x_i - f_{\text{nonlinear}}(\mathbf{x}))_i)^2 \tag{46}$$

where M_i are weights which control the importance of each feature. We choose

$$M_i = \begin{cases} 1 & \text{if } i \leq \frac{n_s}{2} \\ \alpha & \text{if } i > \frac{n_s}{2} \end{cases} \tag{47}$$

for some parameter $\alpha \in [0, 1]$. This breaks the symmetry because some features are now more important than other features. As before we train until the loss function ceases to decrease, with a batch size of 4096 and a learning rate of .0003. We train a model with $n_s = 4096$ sparse features which have the same $p = .04$ of activating. The compressed dimension $n_d = 1024$.

First let us check that despite the symmetry breaking all features are still represented. Looking at the model with $\alpha = \frac{1}{2}$ we can consider the diagonal and off-diagonal terms. The diagonal terms can be broken up into two groups: W_{ii} for $i \leq n_s/2$ which have high importance, and W_{ii} for $i > n_s/2$ which have lower importance. We plot a histogram of these terms in the first panel of fig. 8. The terms within each group are close to uniform whereas the two groups have somewhat different means with the important features having slightly larger diagonal entries.

The off-diagonal terms can be summarized by the standard deviations of ν_i as before, again split into more and less important groups. We see that the variance of ν_i is almost uniform inside each group, and smaller for the more important features. This shows that permutation symmetry is maintained

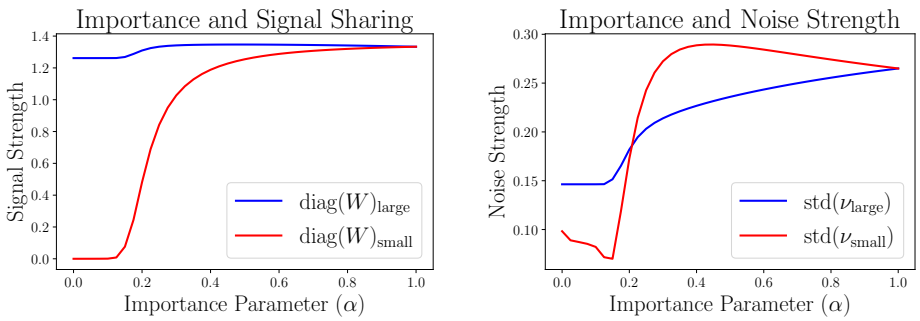

Figure 9: This figure shows how the signal strength (left) and noise level (right) are shared between the more and less important features as the importance parameter α goes from 0 to 1. The signal strength is given by the mean diagonal value of W, while the noise is given by the mean over rows of the standard deviation of ν_i. When α is small all the capacity of the model is directed towards the more important features. As α increases the model begins to dedicate some capacity towards the less important features. At this point the model pushes more of the noise towards the less important features. The model breaks symmetry smoothly near $\alpha = 1$.

within each group, all features may still be represented, and hints that the model gracefully deviates from our permutation-symmetric solution by shifting it's budget for the noise and signal.

To see that this shift behaves nicely as α varies away from 1 (which recovers full permutation symmetry), we look at the mean diagonal value in each group (left panel of fig. 9), and the mean standard deviation of the noise ν_i (right panel of fig. 9) in each group. As we can see for α near 1 the behavior shown in the histograms is maintained. As α becomes smaller the model initially decreases the signal strength, and increases the noise associated with the less important group of sparse features.

Around $\alpha \approx .25$ (for this set of parameters) the model begins to give up entirely on encoding the less important features, which allows it to increase the fidelity of the more important group. It does this primarily by reducing the noise sent to those features.

Here we see that the deviations from permutation symmetry produce slowly varying changes in the optimal encoding strategy for a wide range of α. This implies that qualitative features of the permutation symmetric setting may remain, even when this symmetry is broken in a more realistic setting.