How Much Information Fits in a Vector?

Anonymous authors
Paper under double-blind review

Abstract

Recent work in neural network interpretability has suggested that hidden activations of some deep models can be viewed as linear projections of much higher-dimensional vectors of sparse latent "features." In general, this kind of representation is known as a superposition code. This work presents an information-theoretic account of superposition codes in a setting applicable to interpretability. We show that when the number k of active features is very small compared to the number N of total features, simple inference methods currently used by sparse autoencoders can reliably decode a d-dimensional superposition code when d is a constant factor greater than the Shannon limit. Specifically, when $\ln k / \ln N \leq \eta < 1$ and H is the entropy of the latent vector in bits, asymptotically it suffices that $d/H > C(\eta)$ for certain increasing functions $C(\eta)$. However, the behavior of $C(\eta)$ depends on what decoding method is used. For example, when $\eta = 0.3$, we empirically show that a method based on the popular top-k activation function typically requires a factor of $C = 4$ dimensions per bit. On the other hand, we exhibit an algorithm that succeeds with less than 2 dimensions per bit and requires only around 3 times as many FLOPs for the same values of (N, d). We hope this work helps connect research in interpretability with perspectives from compressive sensing and information theory.

1 Introduction

If each neuron in a given neural network coded for a "meaningful feature" of its input, we could hope to reverse-engineer the network's behavior on a neuron-by-neuron basis. However, many large models have been found to learn neurons that correlate simultaneously with apparently unrelated features. This phenomenon is known as polysemanticity (Nguyen et al., 2016; Zhang & Wang, 2023; Olah et al., 2020).

The appearance of so-called "polysemantic neurons" is not surprising from a connectionist viewpoint. Since at least the 1980s, proponents of artificial neural networks have argued that these systems may naturally use **distributed representations**—coding schemes where individual features are represented by patterns spread over many units of computation, and conversely where each unit carries information on many features. This term was apparently coined in Rumelhart et al. (1986), Chapter 3. In contrast, a *local* representation

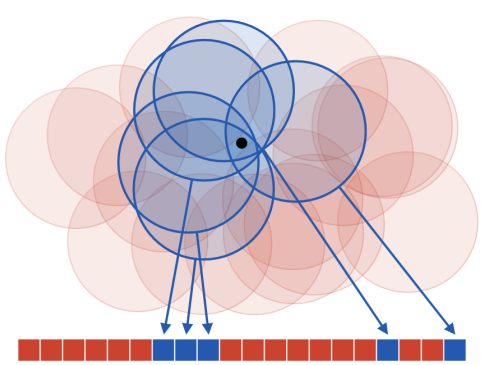

Figure 1: A coarse code representing a point on a plane. Each "neuron," drawn as a red or blue square, encodes whether the point belongs to an associated "receptive field." Although no neuron gives specific information on the position of the point, the overall code determines its position with reasonable accuracy.

would dedicate each unit to a single feature. See Thorpe (1989) for a general discussion of local and distributed codes. Figure 1 illustrates a classic example of a coarse code, one kind of distributed representation.

As of yet, relatively little is known about how large neural networks learn to represent information in their hidden layers or to what extent this information can be interpreted. However, should "interpretable features" exist, the connectionist viewpoint makes it natural that they would be stored with non-local codes. This is a common assumption in interpretability research today; for example, when Meng et al. (2022) intervened on an MLP layer of a language model to "edit" a factual association, both the "subject" and the "fact" were modeled as vectors of activations rather than individual channels.

To infer some kind of latent features from an activation vector y, one simple proposal is to model y as a linear projection $y = Fx$ of a high-dimensional and *sparse* vector x. This idea is based on the remarkable fact that, for certain families of matrices F and for certain constraints on the sparsity of x, a d-dimensional linear image y codes uniquely for an N-dimensional sparse vector x when $N = \Omega(e^d)$.

We refer to the columns of F as codewords and the whole matrix F as a dictionary. Since y is a linear superposition of codewords, it is called a **superposition code** for x. The task of inferring x from a superposition code is known as sparse reconstruction, and the task of inferring the dictionary F from a distribution over the codes y is called dictionary learning. Both of these problems have been studied in the field of compressive sensing, although with different applications in mind; see Elad (2010) for a review of classic work in the context of signal and image processing.

Already in 2015, Faruqui et al. (2015) used a dictionary learning method to derive sparse representations for word embeddings and argued that these latents were more interpretable than the original embedding dimensions. More recently, a series of works beginning with Yun et al. (2021) have applied dictionary learning to the internal representations of transformer-based language models. Cunningham et al. (2023) suggested the use of **sparse autoencoders** (SAEs) and Templeton et al. (2024); Gao et al. (2024) scaled sparse autoencoders to production-size large language models.

Templeton et al. (2024) showed that latent features learned by SAEs are often highly interpretable, and that intervention on these features allows "steering" language models in predictable ways. However, as reported in Gao et al. (2024), even SAEs with extremely large numbers of latents suffer from an apparently irreducible reconstruction error. According to Sharkey et al. (2025), understanding the limitations of SAEs—and dictionary learning in general—is an important open question in the research program of mechanistic interpretability.

1.1 Contributions

To infer a latent representation $x \in \mathbb{R}^N$ from an activation vector $y \in \mathbb{R}^d$, sparse autoencoders use a simple estimate of the form $\hat{x}(y) = \sigma(Gy)$ where $G \in \mathbb{R}^{N \times d}$ is a learnable matrix and σ is some kind of thresholding operation. Throughout this paper, we refer to any estimate of this form as a "one-step estimate" for x, since it is not an iterative method. Gao et al. (2024), a work representative of research on large-scale sparse autoencoders, considered latent vectors with dimension N on the order of 2^{20} and number of non-zero coefficients k in each latent vector $\hat{x}$ around 64.

Research on sparse autoencoders is mired by many kinds of scientific uncertainty. Of course, it is not known a priori how well activation vectors can be effectively modeled as sparse superposition codes. It is not even necessarily clear what is meant by an "interpretable feature," as per Zytek et al. (2022). Furthermore, even if some activity of a model can be modeled as a superposition code, it's not obvious whether the latent features being encoded can be recovered by a one-step estimate. Indeed, the compressive sensing literature provides many iterative methods for sparse recovery that require more computation but succeed more reliably. To what extent are the limitations of SAEs explained by the limitations of our sparse recovery methods?

One way to frame this question is to ask *how much information* a superposition code may hold. Classically, this is addressed by the tools of information and coding theory. Given a "channel" with certain characteristics—for example, a band-limited telephone connection or a binary storage device with a certain failure rate—the amount of information we can encode is asymptotically characterized by a channel "capacity" measured in bits per unit of channel usage.

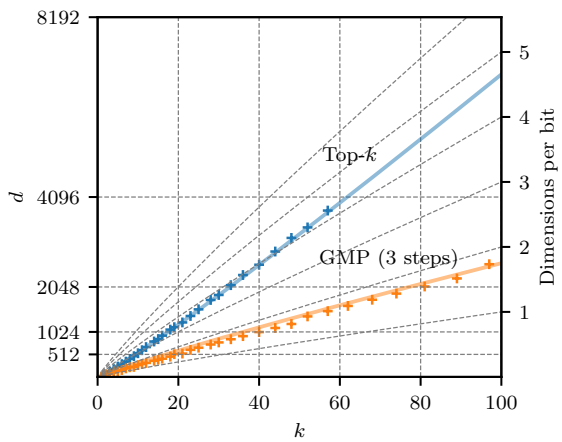

Figure 2: Crosses give minimum embedding dimensions d at which two different methods were empirically found to completely decode a superposition code for a k-sparse subset of $\{1, \ldots, 2^{20}\}$ with probability 0.95 and colored lines give rules of thumb derived from our theoretical discussion in Section 4. The inverse "bitrate" $d/\tilde{H}_2$, where $\tilde{H}_2 = k \log_2(eN/k) \approx \log_2 \binom{N}{k}$, is indicated by the right axis. Top-k is a method currently used by sparse autoencoders, and generalized matching pursuit (GMP) is a simple algorithm described in Section 2.

The goal of this paper is to perform a similar analysis for sparse superposition codes in a setting that is meaningful to sparse autoencoders. More specifically, we consider a situation where the latent vector x is extremely sparse and must be recoverable from the code y by a relatively efficient method. Our analysis focuses on a specific toy example described in Section 2. Our main contributions are the following.

1. Asymptotically in a regime of sublinear sparsity (meaning that $k \leq N^\gamma$ for some $\gamma < 1$), we prove that one-step estimates used by sparse autoencoders can reliably decode superposition codes $y \in \mathbb{R}^d$ so long as d is a constant factor greater than the Shannon limit (Proposition 4). Furthermore, our results lead to accurate predictions for the empirical performance of the popular top-k method.

2. We investigate a simple extension of top-k which we call **generalized matching pursuit** (GMP). By numerical experiment, we show that GMP can decode superposition codes carrying more than two times as much information while requiring, for the same values of (N, d, k), only a small constant factor more computation than top-k.

A concrete result of our findings is illustrated in Figure 2. For $k \in \{1, \ldots, 100\}$ we show the number of dimensions d required for two different methods to accurately read a k-sparse subset of $\{1, ..., 2^{20}\}$ that has been transmitted as a d-dimensional superposition. (The dictionary F is randomly generated; see Section 2 for more details.) As k increases, the ratio d/H of required dimensions per bit of entropy increases when we use a simple decoding method based on the top-k activation function. For moderate values of k, this method requires more than 4 dimensions per bit. In particular, a superposition code with more than 60 elements requires an embedding dimension greater than 4096. In contrast, generalized matching pursuit (GMP) with $T = 3$ steps requires less than two dimensions per bit, while requiring only around 3 times more FLOPs for the same values of (N, d, k).

1.2 Relation to Other Work

The basic problem of recovering a sparse vector x from a linear projection is studied in compressive sensing. This field stems from works such as Donoho (2006) and Candes & Tao (2005), which showed that systems of linear measurements can be used to reconstruct certain kinds of sparse signals even when the number d of linear measurements is much smaller than the nominal dimension N of the signal. This was an important observation in signal processing, since many real-world signals are known to be approximately "sparse" in some appropriate basis. For example, images often admit sparse approximations in certain wavelet bases; see Chapter 10 of Elad (2010).

A sparse signal can naturally be encoded (either exactly or approximately) with only a small fraction of the information that would be required to store each of its N dimensions explicitly. The surprising result of

compressive sensing is that compression can be performed by a system of *non-adaptive* linear measurements—that is, measurements (or "sensors") which are fixed a priori—in such a way that decompression can be performed in practice. This is the explanation for the term compressive (or compressed) sensing. On the other hand, the subsequent operation needed to reconstruct the sparse signal x from the vector of linear measurements y is necessarily non-linear. Two important theoretical models for this operation are the solution of the linear program

$$\textbf{minimize } |x|_1 \textbf{ subject to } Fx = y$$

and the algorithm of approximate message passing (Donoho et al., 2009).

In compressive sensing, the matrix $F \in \mathbb{R}^{d \times N}$ is often called the design matrix. For linear projections Fx to faithfully encode sparse vectors x, it is intuitive that the rows of the design matrix must not be too aligned with the basis vectors of $\mathbb{R}^N$. More precisely, the *nullspace property* (Cohen et al., 2009) formalizes the idea that the nullspace of F must not contain elements which are "too sparse." Remarkably, it can be shown that design matrices with independent and randomly sampled entries have good compressive sensing properties.

The idea of using measurements with randomly chosen coefficients bears a resemblance to the idea that, in a distributed representation, each unit of storage should integrate over many different "features" of the information being stored. (Such measurements are also called *holographic*, and are reminiscent of the operation of optical holography.) Connections between compressive sensing and vector representations used in deep learning have been identified before, for example in Arora et al. (2015).

Compressive sensing is a deep field of study and has received considerable attention over the last two decades. Given a model for the underlying signal and the system of measurements, a typical problem is to characterize the minimum number of measurements d required for a certain reconstruction method to achieve some standard of performance. This minimal value is called a sample complexity. For example, Reeves et al. (2019) studied the sample complexity of the maximum likelihood estimator in a regime of sublinear sparsity and in the presence of a small amount of measurement noise. Many practical methods for sparse reconstruction are available, and more continue to be developed; for example, Takeuchi (2024) extended the approximate message passing algorithm to achieve better sample complexity for signals with sublinear sparsity.

However, many results of compressive sensing are not immediately relevant to the study of sparse autoencoders in interpretability. First, in compressive sensing it is common to consider a regime of *linear sparsity*, where $k \geq \epsilon N$ for some $\epsilon \gg 0$, whereas in latent representations found by SAEs the ratio k/N is typically extremely small. Furthermore, even in works that consider sublinear sparsity, decoding methods are typically either iterative (Truong, 2023; Takeuchi, 2024) or require specially designed dictionaries (Li et al., 2019). Although prior works like Bajwa et al. (2010) have studied "one-step" methods for sparse reconstruction, we were not able to find an analysis of these methods in a regime meaningful for SAEs. For example, it was not clear how to model the sample complexity of a top-k autoencoder. The present work serves as an analytical starting point on this topic.

1.3 Structure of This Work

The following sections are structured as follows.

- Section 2 introduces the toy superposition code to be studied and the decoding methods we will consider. These are MAP (maximum a posteriori) threshold decoding, top-k decoding, and GMP (generalized matching pursuit).
- Section 3 briefly recalls the idea of a Shannon limit and explains some heuristic predictions for sample complexity.
- Section 4 gives our theoretical results on the performance of threshold and top-k decoding as well as our numerical comparison with GMP.
- Finally, Section 5 discusses our choice to use random dictionaries.

2 Superposition Codes and Sparse Recovery

We begin by describing the toy scenario to be studied. Given a large number N, consider a map F that represents each subset $x \subseteq [N] = \{1, \ldots, N\}$ by a linear combination

$$y(x) = \sum_{i \in x} f_i \in \mathbb{R}^d,$$

where the vectors $\{f_i \in \mathbb{R}^d : i \in [N]\}$ are chosen in advance and where the dimension d of the encoding is expected to be much smaller than N. The vectors f_i are called codewords for the elements of $[N]$, the collection $\{f_1, \ldots, f_N\}$ is called a dictionary, and the image Fx is called a superposition code. By viewing x as a vector in $\{0,1\}^N$ with coefficients

$$x_i = \begin{cases} 1 : i \in x \\ 0 : \text{otherwise} \end{cases}$$

and viewing F as the matrix of column vectors $[f_1 \ \ldots \ f_N]$, we can write $y = Fx$ as a linear equation. In this work, we'll model our subset as a random variable X uniformly distributed over all subsets of some fixed size $k \ll N$. Keeping with the orders of magnitude discussed in Gao et al. (2024), we consider $N \approx 2^{20}$ and $k \approx 64$.

The problem of recovering a sparse signal x from a linear projection Fx is known as sparse recovery. In this work, we will use language from coding theory and think of x as some data to be encoded and the projection $y = Fx$ as a code. Thus, sparse recovery becomes the task of "decoding" x. Note that this is opposite to the convention used in sparse autoencoders, where the map estimating x from y is called the "encoder." (In classical applications of autoencoders, the learned representation typically is understood as an efficient code for the data distribution and has fewer dimensions than the data being modeled.)

Due to a connection with sparse linear regression, the subproblem of identifying the support of x from the projection Fx is often called model selection. When the coordinates of x take values in $\{0,1\}$, sparse recovery and model selection coincide. In general, note that model selection presents the main difficulty of sparse recovery; once the support of x is identified, its non-zero coordinates can be estimated easily e.g. by solving a least squares problem.

In practice, a dictionary F employed by a neural network will likely have special properties related to the nature of the data being encoded and the operations that need to be performed. However, for the purposes of encoding a uniformly random subset of $[N]$, one natural way to choose a dictionary is to make each codeword an independent random vector. The simplest motivation for this strategy is the fact that, so long as $d = \Omega(\ln N)$, a dictionary of N randomly chosen vectors is highly "incoherent." For example, we can prove the following for dictionaries of Rademacher vectors.

Proposition 1. *For $\mu > 0$ and $p > 0$, let $d > 2\mu^{-2}(2 \ln N + \ln p^{-1})$, and let $\{F_1, \ldots, F_N\}$ be independent draws from $\{-1/\sqrt{d}, 1/\sqrt{d}\}^d$. Then*

$$\forall i \neq j, \ |\langle F_i, F_j \rangle| < \mu \tag{1}$$

with probability at least $(1-p)$.

For a standard proof, see Appendix C.

The infimum of all constants μ satisfying the condition (1) is called the mutual coherence of a dictionary. Bounds on mutual coherence give a relatively simple way to guarantee the success of various sparse recovery algorithms; for example, see Chapter 4 of Elad (2010). In our scenario, we can say the following.

Remark 1. *Given a dictionary F with mutual coherence smaller than $1/2k$, for any k-sparse vector $x \in \{0,1\}^N$ and for all $i = 1, \ldots, N$ we have*

$$\forall i, x_i = \begin{cases} 1 : \langle f_i, Fx \rangle \geq 1/2 \\ 0 : \textit{otherwise.} \end{cases}$$

Altogether we find that, for a fixed k, it suffices to take $d = \Omega(\ln N)$ in order for each coefficient x_i to be retrievable in a very simple way. In the remainder of this work, we are interested in refining this prediction, and especially in understanding how the minimum reliable value of d depends on our decoding method.

We consider two natural families of random dictionaries. A **Rademacher dictionary** is a random dictionary with codewords drawn as in Proposition 1, i.e. independent Rademacher vectors normalized to have unit norm. A **spherical dictionary** is a random dictionary with codewords drawn uniformly and independently from the unit sphere. These dictionaries have similar incoherence properties in our regime of interest; a result we prove for one dictionary (such as Proposition 1) can typically be proven in a similar form for the other.

Finally, it remains to define the decoding methods we will study. We informally refer to decoding methods that require only a single "step" of estimation and thresholding, like the estimate in Remark 1, as one-step decoders. In this work, we will consider two one-step methods and one very simple iterative algorithm which we call generalized matching pursuit (GMP).

2.1 Threshold Decoding and Top-k Decoding

To motivate one-step decoders, consider the problem of estimating a single coefficient X_i from the model

$$Y = X_i f_i + \overbrace{\sum_{j \neq i} X_j f_j}^{Z_i}. \tag{2}$$

(Throughout, our convention will be to write random variables with capital letters.) If we model the sum Z_i as centered Gaussian noise with covariance Σ, the inference problem for X_i becomes tractable; specifically, if we define an inner product by $\langle v, w \rangle_\Sigma = \frac{1}{2} v^T \Sigma^{-1} w$, then the posterior on X_i can be expressed in terms of the linear function

$$\lambda_i(Y) = \frac{\langle f_i, Y \rangle_\Sigma}{\|f\|_\Sigma^2}.$$

In signal processing, estimating a scalar in the presence of noise is called filtering, and this kind of optimal linear measurement of a Gaussian model is called a matched filter. In terms of our matched filter,

$$\ln \mathrm{P}(X_i = x | Y = y) = -\frac{\rho_i}{2} \left(x - \lambda_i(y)\right)^2 + \ln \mathrm{P}(X_i = x) + C \tag{3}$$

where $\rho_i = (\operatorname{Var} \lambda_i(Z_i))^{-1}$ is called the signal-to-noise ratio (SNR). (See Appendix A for a review.) When F is a random dictionary of unit-norm codewords, we can model Z_i as an isotropic Gaussian, in which case our matched filters take the form $\lambda_i(Y) = \langle f_i, Y \rangle$.

One simple decoding strategy would be to base our estimates $\hat{X}_i(Y) \approx X_i$ directly on the matched filters $\langle f_i, Y \rangle$. Specifically, we may put $\hat{X}_i = 1$ if $\langle f_i, Y \rangle$ is larger than some threshold τ and $\hat{X}_i = 0$ otherwise. We call this method **threshold decoding** at level τ (Algorithm 1). Substituting the prior

$$\mathrm{P}(X_i = 0) = \frac{N-k}{N}, \quad \mathrm{P}(X_i = 1) = \frac{k}{N}$$

into our expression (3) for the posterior of the Gaussian model shows that the maximum a posteriori estimate for X_i is equivalent to thresholding at level

$$\tau = \frac{1}{2} - \rho_i^{-1} \ln \frac{\mathrm{P}(X_i = 1)}{\mathrm{P}(X_i = 0)} = \frac{1}{2} - \rho_i^{-1} \ln \frac{k}{N-k}. \tag{4}$$

Using the fact that $\operatorname{Var} \langle F_i, F_j \rangle = 1/d$ for a pair of independent codewords drawn from either a Rademacher or spherical dictionary, it's easy to show that $\operatorname{Var} \langle f_i, Y \rangle = (N-1)k/Nd$. Plugging this expression in as the

variance $\operatorname{Var} \lambda_i(Z_i)$ in our fictitious Gaussian model gives the thresholding level

$$\begin{aligned}\tau_{\text{MAP}} &= \frac{1}{2} - \frac{N-1}{N}\frac{k}{d}\ln\frac{k}{N-k} \\ &= \frac{1}{2} - \frac{N-1}{N}\frac{k}{d}\left(\ln\frac{k}{N} + \frac{k}{N} + O\left(\frac{k^2}{N^2}\right)\right).\end{aligned}$$

We refer to threshold decoding using this value of τ as MAP threshold decoding.

Another very simple strategy is to first compute all the matched filters $F^T Y$ and find which are largest. We will call this top-k decoding (Algorithm 2). Top-k has the disadvantage that k must be known exactly in advance, whereas threshold decoding only requires k to adjust the threshold τ. However, should k be known, it is clear that top-k is strictly more reliable than threshold decoding for any τ.

Input: $y \in \mathbb{R}^d$, $F \in \mathbb{R}^{d\times N}$, $\tau \in \mathbb{R}$
Output: $\hat{x} \in \{0,1\}^N$
$\hat{x} \leftarrow 0$;
$\hat{x} \leftarrow 1$ where $F^T y \geq \tau$;
return $\hat{x}$

Algorithm 1: Threshold decoding

Input: $y \in \mathbb{R}^d$, $F \in \mathbb{R}^{d\times N}$, $k \in \mathbb{N}$
Output: $\hat{x} \in \{0,1\}^N$
$\hat{x} \leftarrow 0$;
$\hat{x} \leftarrow 1$ at k largest values of $F^T y$;
return $\hat{x}$

Algorithm 2: Top-k decoding

Note that in some works (Elad, 2010) top-k is known as the thresholding algorithm. Finally, note that in practice the latent quantities X_i to be estimated take values besides 0 and 1. The two algorithms described here can be viewed as simplifications of one-step methods currently used by autoencoders. Sparse autoencoders using top-k activations were first suggested by Makhzani & Frey (2014).

2.2 Generalized Matching Pursuit

As discussed earlier, the field of compressive sensing provides various iterative algorithms for sparse recovery. For example, one classic starting point is to apply proximal gradient descent to the objective

$$\frac{1}{2d}\|Fx - y\|_2^2 + \frac{\lambda}{d}\|x\|_1$$

where the parameter $\lambda > 0$ can be understood as a Lagrange multiplier. This gives the iterative soft thresholding algorithm (ISTA) (Daubechies et al., 2003), and is the basis for methods like FISTA (Beck & Teboulle, 2009). A single iteration of ISTA involves a matrix-vector product followed by a thresholding operation, and so is computationally comparable to top-k or MAP thresholding. Unrolling ISTA and optimizing its parameters, as we would do for a deep neural network, is called learned ISTA (LISTA) (Gregor & LeCun, 2010). Another notable family of methods, which are not derived from a linear programming objective but instead from a message passing formulation, are based on the approximate message passing algorithm (Maleki, 2010). However, for sparse autoencoders, a training run using top-k is already very computationally expensive. Therefore, it is interesting to study the properties of relatively cheap modifications to top-k.

We propose the following generalization of the top-k decoder, which we call generalized matching pursuit.

Input: Code $y \in \mathbb{R}^d$, dictionary $F \in \mathbb{R}^{d\times N}$, sparsity k, steps T
Output: Latent $\hat{x} \in \{0,1\}^N$
Choose any step sizes $k_1, \ldots, k_T$ so that $\sum_t k_t = k$ and $\max_t k_t$ is minimized ;
$\hat{x} \leftarrow \mathbf{0}$;
for $t = 1, \ldots, T$ **do**
  $\hat{x} \leftarrow 1$ at k_t largest values of $F^T(y - F\hat{x})$;
end
return $\hat{x}$

Algorithm 3: Generalized matching pursuit (GMP) with T steps

This algorithm can also be understood a simplification of generalized orthogonal matching pursuit (Wang et al., 2012). Since the most computationally expensive step of Algorithm 3 is to compute the product $F^T(y - F\hat{x})$, while the vector $r = y - F\hat{x}$ can be efficiently updated from one iteration to the next, GMP with T steps has roughly T times the computational requirements of top-k. As we will see in our results in Section 4, $T = 3$ will be enough to significantly improve over the performance of top-k.

3 Information Theory Bounds

To understand the minimum dimension d required for a superposition code with parameters (N, k) to be decodable by a given method, it will be helpful to use the point of view of information theory.

Let us first consider the case $k = 1$, meaning that the latent variable X to be encoded is just a symbol drawn from an alphabet of size N. What is the minimum dimension d for which each value of X can be represented by a vector $Y \in \mathbb{R}^d$? When the vector Y is stored exactly, the answer to our question simply depends on the number of values each coefficient can take. If the coefficients of Y are 16 bit floating points, then each coefficient can store (nearly) 2^{16} distinct numbers, and overall we need only about $d_{\min} \approx \frac{1}{16} \log_2 N$ dimensions. Applying the same reasoning to the case where X is a k-sparse subset suggests we may need only $\frac{1}{16} \log_2 \binom{N}{k}$ dimensions. However, this is a very optimistic prediction; in practice the ratio $d/\log_2 \binom{N}{k}$ will need to be significantly larger than $1/16$.

In the presence of exogenous noise, information theory provides a simple explanation for this kind of limitation. Specifically, suppose $Z \in \mathbb{R}^d$ is a vector of independent Gaussians each of variance P, and consider the problem of recovering X from $Y + Z$. The following is then a very well-known (but remarkable) result of information theory.

Proposition 2 (A Shannon limit)**.** *Let the random variable $X \in [N]$ be uniformly distributed and let $Z \in \mathbb{R}^d$ be independent Gaussian noise of variance P. Suppose there exists a pair of maps $F\colon [N] \to \mathbb{R}^d$ and $G\colon \mathbb{R}^d \to [N]$ so that*

$$G(F(X) + Z) = X$$

with probability at least $(1 - p)$, and suppose the variance of each coordinate of $F(X)$ is bounded by 1. Then

$$d \geq 2\frac{(1-p)\ln N - \ln 2}{\ln\big(1 + P^{-1}\big)}.$$

Asymptotically for large N, this means that

$$\frac{d}{H} \geq (1 + o(1))\frac{2}{\ln(1 + P^{-1})}$$

if there is any pair (F, G) that transmits X with probability $1 - o(1)$, where $H = \ln N$ is the entropy of X. In other words, to transmit information reliably in the presence of noise of power P, we need at least $2/\ln(1 + P^{-1})$ dimensions per nat.

Conversely, it can be shown that this bound on the ratio d/H is asymptotically tight. More specifically, for any $\epsilon > 0$ and for sufficiently large N, it suffices that

$$\frac{d}{H} \geq (1 + \epsilon)\frac{2}{\ln(1 + P^{-1})}$$

for there to exist some pair (F, G) satisfying the conditions of Proposition 2 with p arbitrarily close to 0. The same conclusions hold more generally under different models for noisy transmission, substituting P^{-1} in general for a "signal-to-noise ratio" (SNR). For a reference on these topics, see e.g. Chapter 9 of Cover & Thomas (2006).

Although our toy example does not involve exogenous noise, it will be useful to understand the sample complexity d of a superposition code in terms of the asymptotically attainable "channel capacity" d/H, where $H = \ln \binom{N}{k}$ is the entropy of the set X being encoded. Before proceeding to our main results, let us see how some lower bounds on d can be informally explained using Shannon limits for noisy channels.

First, let us divide the vector $X \in \mathbb{R}^N$ into two parts (X^0, X^1) each of length $N/2$ and with $k/2$ non-zero entries, and likewise divide the dictionary F into two dictionaries (F^0, F^1) so that $Y = F^0X^0 + F^1X^1$. Now, crucially, suppose we are using a decoder whose estimate for X^0 is independent of F^1, and vice versa. If our decoder is reliable for a randomly constructed dictionary, then it follows that the restriction of our decoder to X^0 is reliable under a noisy channel with signal-to-noise ratio of 1. The inverse capacity of such a channel is $2/\ln(1+1) = 2/\ln 2$, and so

$$d \geq (1 + o(1))\frac{2}{\ln 2} \ln \binom{N/2}{k/2}.$$

Since overall the information to be transmitted is $H = 2\ln\binom{N/2}{k/2}$, we conclude

$$\frac{d}{H} \geq (1 + o(1))\frac{1}{\ln 2},$$

or $d/H_2 \geq 1 + o(1)$ where $H_2 = H/\ln 2$ is the entropy measured in bits. Very roughly, this shows that if a decoder can be separated into two pieces that almost do not communicate, then it cannot read more than 1 bit per dimension from a superposition code. Intuitively, such decoupling properties should hold in approximate senses for many simple decoding methods.

On the other hand, if our estimate $\hat{X}_i$ for each coordinate X_i depends on variables which are statistically independent from *all* other codewords $\{F_j\}_{i \neq j}$—as is the case in threshold decoding—then these estimates are reliable under noise with a signal-to-noise ratio of nearly $1/k$. But for it to be possible to transmit even a single codeword on a channel with SNR of $1/k$, information theory tells us that for large k we need

$$\frac{d}{H} \geq (1 + o(1))\frac{2}{\ln(1 + k^{-1})} \approx (2 + o(1))k,$$

where $H = \ln N$ is the entropy of our choice of codeword. Roughly, this shows that we need $2k \ln N$ dimensions for threshold decoding to be reliable.

4 Main Results

We now return to the toy scenario described in Section 2. For a given family of dictionaries $F_{N,d} \in \mathbb{R}^{d \times N}$, consider the problem of recovering a random k-element subset X from a d-dimensional superposition code. We will write

$$b(d, k, N; \tau), \quad b(d, k, N; \text{top-}k)$$

for respective failure probabilities of threshold decoding at level τ and top-k decoding.[1] Naturally we have $b(d, k, N; \text{top-}k) \leq b(d, k, N; \tau)$, since if a given code Y is accurately decoded by threshold decoding at any level, it is also decoded by top-k.

Our first result provides an asymptotic constraint on the dimension d required for either of these values to converge to 0 for large N.

Proposition 3. *Let $\epsilon > 0$ be arbitrary. Over the regime*

$$d \leq (1 - \epsilon)2k \ln N, \quad 2 \leq k < N/2,$$

for random spherical dictionaries it holds that

$$\lim_{N \to \infty} b(d, k, N; \textit{top-}k) = 1.$$

This proposition is stated for spherical dictionaries, but similar results can naturally be proven for other kinds of random dictionaries. Also note that this result agrees with the informal derivation based on a Shannon limit that we described in the previous section. In fact, it is likely possible to prove Proposition 3

[1] The letter b is a mnemonic for "bad event."

by a refinement of such an information-theoretic argument. However, our proof (in Appendix D) simply uses concentration inequalities to provide adequate lower bounds on the quantity $b(d, k, N; \text{top-}k)$.

Now, in a regime when $k \geq \epsilon N$ for some $\epsilon > 0$, the entropy of the variable X being transmitted is

$$H = \ln \binom{N}{k} \leq k \ln \left(\frac{eN}{k} \right) = O(N).$$

However, Proposition 3 shows that we need at least $d = \Omega(k \ln N) = \Omega(N \ln N)$ dimensions for top-k to read a superposition code. In this regime, often called a regime of *linear sparsity*, we can informally say that the amount of information that top-k can read from each dimension of the code converges to 0.

Is it natural to ask whether top-k can be "information-efficient" in some sparser regime. By our discussion, we would need $H/(k \ln N)$ to be bounded strictly above 0. Since

$$\frac{H}{k \ln N} = \frac{k \ln N - k \ln k + O(k)}{k \ln N} = 1 - \frac{\ln k}{\ln N} + O\left(\frac{1}{\ln N} \right),$$

we conclude that top-k can only be information-efficient in a regime where $\ln k / \ln N$ is bounded strictly below 1. This motives the introduction of a parameter $\eta = \ln k / \ln N$, which we call the sparsity exponent. In compressive sensing, when $k \leq N^\eta$ for some fixed $\eta < 1$ we say we are in a regime of *sublinear sparsity*. Our next result gives a sufficient condition for threshold decoding to succeed asymptotically in such a regime.

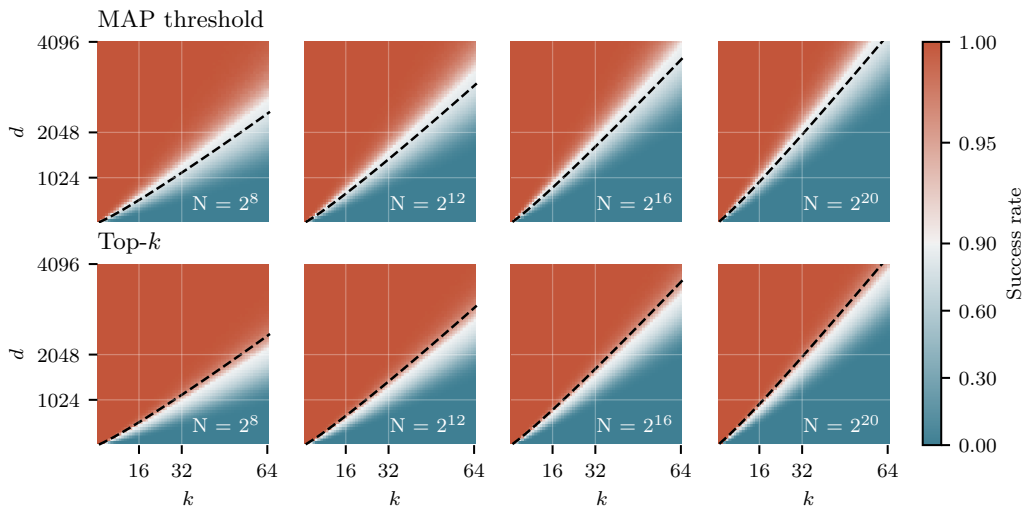

Figure 3: Empirical performance of MAP thresholding and top-k thresholding at decoding a k-element subset of $[N]$ from a d-dimensional superposition code using a random spherical dictionary. The dashed line shows the expression $d = 2(1 + \sqrt{\eta})^2 k \ln N$ appearing in Proposition 4, evaluated with $\eta = \ln k / \ln N$.

Proposition 4. *For $\eta \in (0, 1]$, let $k \leq N^\eta$. Then for any $\epsilon > 0$, over the regime*

$$d \geq (1 + \epsilon) 2 (1 + \sqrt{\eta})^2 k \ln N,$$

for Rademacher dictionaries it holds that

$$\lim_{N \to \infty} b(d, k, N; \tau(\eta)) = 0,$$

where $\tau(\eta) = (1 + \sqrt{\eta})^{-1}$.

Note that expression for $\tau(\eta)$ used here is connected (but not identical) to the MAP threshold we derived in Section 2. A simple extension of Proposition 4 would prove that MAP thresholding has the same asymptotic

guarantee. See Appendix D for proofs of Proposition 3 and Proposition 4, as well as a discussion of this connection.

In the extreme case where $\eta = 1$, the condition $k \leq N^\eta$ is vacuous and overall Proposition 4 lets us conclude that $d \geq (1+\epsilon)8k \ln N$ dimensions are enough for threshold decoding to succeed with threshold $\tau = 1/2$ for any k as $N \to \infty$. For smaller values of η, Proposition 4 shows that we can improve the constant factor by choosing a threshold τ strictly larger than 1/2.

Figure 3 shows how the critical value $d_{\text{crit}} = 2(1+\sqrt{\eta})^2 k \ln N$ appearing in the statement of Proposition 4 compares with empirical performance of MAP thresholding and top-k for finite values of N and k. Note that the statement of our result is asymptotic, and does not give guarantees on the performance of either decoding method in practice for finite k and N. (More explicit statements can be derived from the bound appearing in Lemma 1 of Appendix D.) However, d_{crit} serves empirically as a good rule of thumb; for $k < 64$ and for values of N ranging over several orders of magnitude, d_{crit} can be used to reliably predict the performance of both MAP thresholding and top-k. In particular, although we find top-k is noticeably more reliable, the sample complexities of both methods are similar and grow as a function of k and N in a similar way.

Now we return to the information-theoretic point of view. Using the fact that

$$\frac{H}{k \ln N} = 1 - \eta + O\left(\frac{1}{\ln N}\right),$$

we can restate Proposition 4 in the following way, which is asymptotically equivalent for $\eta < 1$.

Corollary 1. *For $\eta \in (0,1)$, let $k \leq N^\eta$ and let $H = \ln \binom{N}{k}$. Then for any ϵ, over the regime*

$$\frac{d}{H} \geq (1+\epsilon)\frac{2(1+\sqrt{\eta})}{1-\sqrt{\eta}} \tag{5}$$

we have the same conclusion as Proposition 4.

For a given dictionary and a given decoder, let $C(\eta)$ be infimum of all constants C such that $d/H \geq C$ implies the failure rate $b(d,k,N)$ will converge to 0 for large N and k satisfying $k \leq N^\eta$. Informally, $C(\eta)$ is the number of dimensions we need to allocate for each nat of information to be transmitted. Proposition 3 proves that, for spherical dictionaries and top-k decoding,

$$\frac{2}{1-\eta} \leq C(\eta).$$

Meanwhile, Corollary 1 proves that, for Rademacher dictionaries and threshold decoding,

$$C(\eta) \leq \frac{2(1+\sqrt{\eta})}{1-\sqrt{\eta}} = \frac{2(1+\sqrt{\eta})^2}{1-\eta}.$$

In the limit $\eta \to 0$ note that these two expressions match; we find that 2 dimensions per nat (around 1.4 dimensions per bit) is sufficient in the context of Corollary 1 and necessary in the context of Proposition 3. For $\eta > 0$, the expression of Corollary 1 is larger by a factor of $(1+\sqrt{\eta})^2$.

Our analysis leaves open several natural questions about the functions $C(\eta)$. For example, it would be interesting to exactly characterize the functions associated with both MAP thresholding and top-k, investigate whether they are independent of the distribution of random dictionary used, and so on. However, the main goal of this work is to provide a useful analytical starting point for research in interpretability. For this end, it suffices to show how a condition of the form $d/H \geq C(\eta)$ can appear. In the remainder of this section, we compare our theoretical predictions with numerical experiments on MAP thresholding, top-k, and GMP decoding. Our main conclusions are that the empirical minimum values of d/H for both MAP threshold and top-k can be reliably predicted in terms of the expression $d_{\text{crit}} = 2(1+\sqrt{\eta})^2 k \ln N$ appearing in Proposition 4, while GMP decoding with only $T = 3$ steps achieves significantly better rates.

When $k \leq N^\eta$ for $\eta < 1/2$, it will be helpful to know that a good approximation for the entropy $H = \ln \binom{N}{k}$ is given by

$$\tilde{H} = k \ln(eN/k) = \ln \binom{N}{k} + O(N^{2\eta - 1}).$$

(See Appendix B for a derivation.) For example, when $N = 2^{16}$ and $k \leq 5000$, the relative error of this approximation is smaller than 1.4%. Let us also write $\tilde{H}_2 = \tilde{H}/\ln 2$ for the expression of $\tilde{H}$ in bits.

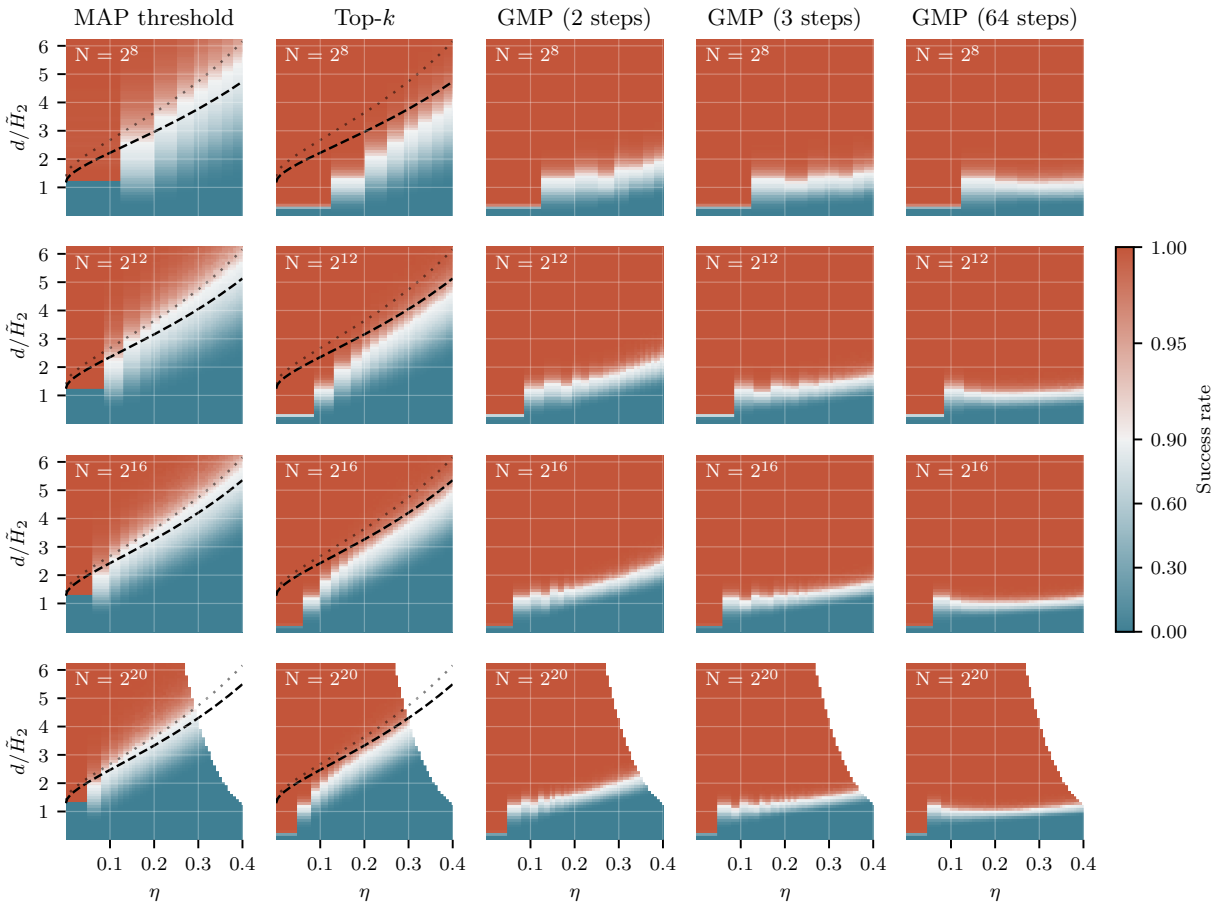

Figure 4: **How many dimensions do we need to store one bit of information?** We graph empirical performance of MAP thresholding, top-k, and GMP at exactly decoding a superposition code as a function of approximate inverse bitrate $d/\tilde{H}_2 = d/(k \log_2(eN/k))$ and sparsity exponent $\eta = \ln k/\ln N$. The bold line shows the expression $d = 2(1+\sqrt{\eta})^2 k \ln N$ occurring in Proposition 4 and thin line shows the limit $d/\tilde{H}_2 = (2\ln 2)(1+\sqrt{\eta})/(1-\sqrt{\eta})$ attained for large N. In all cases, we use a dictionary of random spherical codewords.

Figure 4 shows the empirical performance of various decoding methods in terms of the (approximate) inverse bitrate $d/\tilde{H}_2$ and the exponent $\eta = \ln k/\ln N$, as N ranges over several orders of magnitude and $\eta \in [0, 0.4]$. Results on the bottom row are truncated where they would require a dictionary with dimensions larger than 4096×2^{20}.

First consider the two left-most columns of Figure 4, which show the performance of MAP threshold decoding and top-k. If we take the expression $d_{\text{crit}} = 2(1+\sqrt{\eta})^2 k \ln N$ in Proposition 4 as a "rule of thumb" for finite values of (k, N), we should predict a transition in Figure 4 at

$$\frac{d_{\text{crit}}}{\tilde{H}} = \frac{2(1+\sqrt{\eta})^2}{1-\eta+1/\ln N}.$$

This value is indicated by the bold line in the graph, while its limit $2(1+\sqrt{\eta})/(1-\sqrt{\eta})$ for large N is indicated by a thin line.

Especially for small η, the performance of top-k is significantly better than MAP thresholding. On the other hand, the sample complexity of both methods can be predicted for $\eta > 0.2$ in terms of the value d_{crit}, as we observed before in Figure 3. With the values $N = 2^{20}$ and $k = 64$ we referenced from Gao et al. (2024), $d_{\text{crit}}/\tilde{H}$ has a value of around 4.3 dimensions per bit. This prediction is slightly optimistic for MAP thresholding, but agrees very closely with the empirical sample complexity we find for top-k decoding.

The next three columns of Figure 4 show the performance of generalized matching pursuit (GMP). We find that, even with $T = 3$ steps, GMP succeeds reliably with $d/\tilde{H}_2 > 2$ over the range $\eta \in (0, 0.35)$. In the case $N = 2^{20}$ and $k = 64$, three-step GMP decreases the required dimensions by a factor of more than 2. Increasing the step parameter further shows noticeable but diminishing returns. Overall, we find that a relatively small increase in computation can have outsized effects on sample complexity in a regime of sublinear sparsity.

5 Are Random Dictionaries Optimal?

In the toy scenario studied in this work, the dictionary F is randomly generated. Of course, the dictionaries employed by neural networks may have special structure. It is natural to ask whether our findings would be significantly different if the dictionary were specially designed in some way.

One natural measurement for the coherence of a dictionary is the mean squared interference of codewords

$$\xi(F) = \binom{N}{2}^{-1} \sum_{i<j} \langle f_i, f_j \rangle^2.$$

Indeed, if we assume an isotropic distribution of codewords, from the point of view of Section 2 the mean squared interference controls the average signal-to-noise ratio of the "filters" $\lambda_i(Y) = \langle f_i, Y \rangle$.

An important result in coding theory known as the Welch bound (Welch, 1974) states that, for any dictionary $F \in \mathbb{R}^{d \times N}$ of unit norm codewords, we have

$$\sum_{i,j} \langle f_i, f_j \rangle^2 \geq \frac{N^2}{d}.$$

This can be derived by viewing the sum of squared inner products above as the sum of squared eigenvalues λ_i^2 of the Gram matrix $F^T F$ and applying a Cauchy–Schwartz inequality, using the fact that $\sum_i \lambda_i = N$.

In terms of mean squared interference $\xi(F)$, the Welch bound means that

$$\xi(F) \geq \frac{1}{N^2 - N} \left(\frac{N^2}{d} - N \right) = \frac{1}{d} \left(1 - \frac{d}{N} \right) \left(1 + \frac{1}{N-1} \right).$$

Therefore, in a limit where $N \to \infty$ and $d/N \to 0$, we have $\xi(F) \geq (1 + o(1))/d$. However, it is easy to check that $\mathrm{E}\,\xi(F) = 1/d$ for both Rademacher and spherical dictionaries, and moreover $\xi(F)$ concentrates tightly around this value as N and d grow. So, in such a limit, the minimum value of $\xi(F)$ does not improve meaningfully over the value attained by a random dictionary. It stands as a conjecture that, in the sublinear regime, the performance of MAP thresholding and top-k are not significantly improved by any family of structured dictionaries.

6 Conclusions and Future Work

Over the course of computation, computer programs use memory to record intermediate results. Similarly, residual vectors within a neural network may encode intermediate "features" relevant to the task being carried out. Reverse engineering representations used by large neural networks, like large language models, is a broad goal considered within mechanistic interpretability. (See, for example, Section 2.1 of Sharkey et al. (2025).)

To a certain extent, sparse autoencoders (SAEs) have succeeded at modeling representations used by language models as superposition codes. However, these methods are limited for reasons that are not yet well understood. It is natural to ask whether some of these limitations may be caused by the restrictive methods that current SAEs use to solve the problem of sparse reconstruction.

This work provides an analytical starting point to answer this question. In a toy example designed to emulate the extremely sparse representations modeled by sparse autoencoders, we have compared two "one-step" methods currently in use by SAEs with a simple iterative algorithm (GMP) whose computational cost is larger by only a small factor. Our main conclusion is that the sample complexity of both one-step methods considered is a constant factor larger than the sample complexity of GMP. Asymptotically, the factor in question appears to depend on the ratio $\eta = \ln k / \ln N$. In the case $\eta = 0.3$, as we have when $N = 2^{20}$ and $k = 2^8$, we find as a rule of thumb for large N that top-k decoding requires around 4 dimensions per bit, whereas GMP succeeds with less than 2 dimensions per bit.

Of course, even if the representations used by large language models can be adequately modeled as superposition codes, we would expect both the dictionaries and the latent variables to have properties not considered in this work. However, the "bitrate gap" between top-k and GMP that we found in our toy example can reasonably be expected to extend to other cases.

The most obvious result of our findings is to suggest that the performance of SAEs may be improved by using slightly more computation to solve the sparse reconstruction problem.[2] However, we hope that the information-theoretic point of view described in this work is also useful to research in this field in a broader way.

In general, when we hypothesize that some vector within a neural network acts as a representation for some more interpretable signal, we can ask about its "bitrate"—that is, the ratio between the number of dimensions used for the encoding and the entropy of our statistical model for the latent signal. How many "bits per dimension" might be stored within real neural networks? On the basis of our toy example, this work tentatively suggests that that certain simple decoding methods may underestimate the amount of information stored. In general, we hope the point of view of coding efficiency helps guide the interpretation of neural representations in future work.

References

Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. Simple, Efficient, and Neural Algorithms for Sparse Coding, March 2015. URL `http://arxiv.org/abs/1503.00778`. arXiv:1503.00778 [cs].

Waheed U. Bajwa, Robert Calderbank, and Sina Jafarpour. Why Gabor frames? Two fundamental measures of coherence and their role in model selection. *Journal of Communications and Networks*, 12(4):289–307, 2010. URL `https://ieeexplore.ieee.org/abstract/document/6388466/`. Publisher: KICS.

Keith Ball. An Elementary Introduction to Modern Convex Geometry. In Silvio Levy (ed.), *Flavors of Geometry*, Mathematical Sciences Research Institute Publications, pp. 1–58. Cambridge University Press, Cambridge, 1997. ISBN 978-0-521-62962-1. doi: 10.1017/9781009701853.002. URL `https://www.cambridge.org/core/books/flavors-of-geometry/an-elementary-introduction-to-modern-convex-geometry/09EA8B107A9ECA496914D95520C9EAFB`.

Amir Beck and Marc Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, January 2009. ISSN 1936-4954. doi: 10.1137/080716542. URL `https://epubs.siam.org/doi/10.1137/080716542`.

E.J. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, December 2005. ISSN 1557-9654. doi: 10.1109/TIT.2005.858979. URL `https://ieeexplore.ieee.org/document/1542412`. Conference Name: IEEE Transactions on Information Theory.

[2] Note that the improved "encoder layers" would need to be used during training of an SAE; if sparse reconstruction is solved inaccurately while the dictionary is being optimized, it is natural that many codewords will not be inferred.

Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Compressed sensing and best k-term approximation. *Journal of the American Mathematical Society*, 22(1):211–231, January 2009. ISSN 0894-0347, 1088-6834. doi: 10.1090/S0894-0347-08-00610-3. URL https://www.ams.org/jams/2009-22-01/S0894-0347-08-00610-3/.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition*. Wiley-Interscience, Hoboken, N.J, 2006. ISBN 978-0-471-24195-9.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse Autoencoders Find Highly Interpretable Features in Language Models, October 2023. URL http://arxiv.org/abs/2309.08600. arXiv:2309.08600.

Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, January 2003. ISSN 1042-9832, 1098-2418. doi: 10.1002/rsa.10073. URL https://onlinelibrary.wiley.com/doi/10.1002/rsa.10073.

Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, November 2003. URL http://arxiv.org/abs/math/0307152. arXiv:math/0307152.

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, November 2009. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0909892106. URL https://pnas.org/doi/full/10.1073/pnas.0909892106.

D.L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006. ISSN 0018-9448. doi: 10.1109/TIT.2006.871582. URL http://ieeexplore.ieee.org/document/1614066/.

M. Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer, New York, 2010. ISBN 978-1-4419-7010-7 978-1-4419-7011-4. OCLC: ocn646114450.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith. Sparse Overcomplete Word Vector Representations. In Chengqing Zong and Michael Strube (eds.), *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1491–1500, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1144. URL https://aclanthology.org/P15-1144/.

Simon Foucart and Holger Rauhut. Sparse Recovery with Random Matrices. In Simon Foucart and Holger Rauhut (eds.), *A Mathematical Introduction to Compressive Sensing*, pp. 271–310. Springer, New York, NY, 2013. ISBN 978-0-8176-4948-7. doi: 10.1007/978-0-8176-4948-7_9. URL https://doi.org/10.1007/978-0-8176-4948-7_9.

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, June 2024. URL http://arxiv.org/abs/2406.04093. arXiv:2406.04093 [cs].

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pp. 399–406, Madison, WI, USA, June 2010. Omnipress. ISBN 978-1-60558-907-7.

Xiao Li, Dong Yin, Sameer Pawar, Ramtin Pedarsani, and Kannan Ramchandran. Sub-linear time support recovery for compressed sensing using sparse-graph codes. *IEEE Transactions on Information Theory*, 65 (10):6580–6619, 2019. URL https://ieeexplore.ieee.org/abstract/document/8733902/. Publisher: IEEE.

Alireza Makhzani and Brendan Frey. k-Sparse Autoencoders, March 2014. URL http://arxiv.org/abs/1312.5663. arXiv:1312.5663.

Arian Maleki. *Approximate message passing algorithms for compressed sensing*. PhD Thesis, Stanford University, 2010. URL https://search.proquest.com/openview/6e7184ed82e021cf5880ebf97b61fe87/1?pq-origsite=gscholar&cbl=18750&diss=y.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks, May 2016. URL http://arxiv.org/abs/1602.03616. arXiv:1602.03616 [cs].

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom In: An Introduction to Circuits. *Distill*, 5(3):10.23915/distill.00024.001, March 2020. ISSN 2476-0757. doi: 10.23915/distill.00024.001. URL https://distill.pub/2020/circuits/zoom-in.

Galen Reeves, Jiaming Xu, and Ilias Zadik. The All-or-Nothing Phenomenon in Sparse Linear Regression. In *Proceedings of the Thirty-Second Conference on Learning Theory*, pp. 2652–2663. PMLR, June 2019. URL https://proceedings.mlr.press/v99/reeves19a.html. ISSN: 2640-3498.

David E. Rumelhart, James L. McClelland, and AU. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. The MIT Press, 1986. ISBN 978-0-262-29140-8. doi: 10.7551/mitpress/5236.001.0001.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Tom McGrath. Open Problems in Mechanistic Interpretability, January 2025. URL http://arxiv.org/abs/2501.16496. arXiv:2501.16496 [cs].

Keigo Takeuchi. Generalized Approximate Message-Passing for Compressed Sensing with Sublinear Sparsity, September 2024. URL https://arxiv.org/abs/2409.06320v2.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet. *Transformer Circuits Thread*, May 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Simon Thorpe. Local vs. Distributed Coding. *Intellectica*, 8(2):3–40, 1989. doi: 10.3406/intel.1989.873. URL https://www.persee.fr/doc/intel_0769-4113_1989_num_8_2_873. Publisher: Persée - Portail des revues scientifiques en SHS.

Lan V. Truong. Fundamental limits and algorithms for sparse linear regression with sublinear sparsity. *Journal of Machine Learning Research*, 24(64):1–49, 2023. ISSN 1533-7928. URL http://jmlr.org/papers/v24/21-0543.html.

Jian Wang, Seokbeop Kwon, and Byonghyo Shim. Generalized Orthogonal Matching Pursuit. *IEEE Transactions on Signal Processing*, 60(12):6202–6216, December 2012. ISSN 1941-0476. doi: 10.1109/TSP.2012.2218810. URL https://ieeexplore.ieee.org/document/6302206.

L. Welch. Lower bounds on the maximum cross correlation of signals (Corresp.). *IEEE Transactions on Information Theory*, 20(3):397–399, May 1974. ISSN 1557-9654. doi: 10.1109/TIT.1974.1055219. URL https://ieeexplore.ieee.org/document/1055219.

Zeyu Yun, Yubei Chen, Bruno Olshausen, and Yann LeCun. Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors. In Eneko Agirre, Marianna Apidianaki, and Ivan Vulić (eds.), *Proceedings of Deep Learning Inside Out (DeeLIO): The*

2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pp. 1–10, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.deelio-1.1. URL https://aclanthology.org/2021.deelio-1.1/.

Changwan Zhang and Yue Wang. A sample survey study of poly-semantic neurons in deep CNNs. In *International Conference on Computer Graphics, Artificial Intelligence, and Data Processing (ICCAID 2022)*, volume 12604, pp. 849–855. SPIE, May 2023. doi: 10.1117/12.2674650.

Alexandra Zytek, Ignacio Arnaldo, Dongyu Liu, Laure Berti-Equille, and Kalyan Veeramachaneni. The Need for Interpretable Features: Motivation and Taxonomy. *ACM SIGKDD Explorations Newsletter*, 24 (1):1–13, June 2022. ISSN 1931-0145, 1931-0153. doi: 10.1145/3544903.3544905. URL https://dl.acm.org/doi/10.1145/3544903.3544905.

A Matched Filters

Consider the problem of inferring a scalar S from the sum

$$X = Sf + Z$$

where $f \in \mathbb{R}^n$ is fixed and Z is a Gaussian variable independent from S. Suppose for simplicity that Z has non-singular covariance Σ, so that $-\ln p(z) = 1/2\|z\|_\Sigma^2$ where

$$\|z\|_\Sigma^2 = z^T \Sigma^{-1} z.$$

Then a routine calculation shows that

$$\begin{aligned} &-\ln p(S = s | X = x) \\ &= C(x) - \ln p(s) + \frac{1}{2}\left(s - \frac{\langle f, x\rangle_\Sigma}{\|f\|_\Sigma^2}\right)^2 \|f\|_\Sigma^2 \end{aligned} \tag{6}$$

where $C(x)$ is a constant depending only on x and $\langle -, -\rangle_\Sigma$ is the inner product associated with the norm $\|-\|_\Sigma$. In particular, the distribution of S conditional on X is only a function of the inner product $\langle f, X\rangle_\Sigma$. The **matched filter** for S is the linear function

$$\lambda(x) = \frac{\langle f, x\rangle_\Sigma}{\|f\|_\Sigma^2},$$

and can be understood as providing the maximum likelihood estimate for S.

In general, the quality of a linear filter λ' as an estimate for S can be measured by the signal-to-noise ratio

$$\rho(\lambda') = \frac{(\lambda'(f))^2}{\mathrm{Var}_Z\, \lambda'(Z)}.$$

The maximum possible signal-to-noise ratio is $\|f\|_\Sigma^2$, which is achieved by λ. Given an improper uniform prior on S, (6) shows the posterior on S conditional on X is a Gaussian with mean $\lambda(X)$ and precision $\|f\|_\Sigma^2$.

B Estimates for the Binomial Coefficient

To estimate $\ln \binom{N}{k}$, it is helpful to first remember the elementary inequalities

$$\left(\frac{N}{k}\right)^k \le \binom{N}{k} \le \left(\frac{eN}{k}\right)^k.$$

Taking logarithms gives $k \ln(N/k) \le \ln \binom{N}{k} \le k \ln(eN/k)$, which proves

$$\ln \binom{N}{k} = k \ln(N/k) + O(k).$$

When $k \ll N$, the upper bound $\ln \binom{N}{k} \le k \ln(eN/k)$ is a better approximation; in fact

$$\ln \binom{N}{k} = k \ln(eN/k) + O(k^2/N).$$

In terms of $k = N^\eta$ the remainder is $O(N^{2\eta - 1})$, which vanishes for $\eta < 1/2$.

One heuristic justification for this finer estimate is to consider a random subset $Y \subseteq [N]$ where each element is included independently with probability $s = k/N$. Then, where

$$h(s) = -s \ln s - (1 - s) \ln(1 - s) = -s \ln s + s + O(s^2)$$

is the binary entropy function, the entropy of Y is

$$\begin{aligned} H(Y) &= h(s)N = -sN\ln s + sN + O(s^2N) \\ &= k\ln(eN/k) + O(k^2/N). \end{aligned}$$

We expect that for large N the number of elements in Y concentrates sharply around k, and so $H(Y)$ should be close to $\ln\binom{N}{k}$.

Indeed this can be proven using the Stirling approximation

$$\ln n! = n\ln n - n + (1/2)\ln(2\pi n) + O(n^{-1}).$$

Substituting into the binomial gives

$$\begin{aligned} \ln\binom{N}{k} &= \ln N! - \ln k! - \ln(N-k)! \\ &= k\ln(N/k) + (N-k)\ln(1 + k/(N-k)) + O(1/N), \end{aligned}$$

and finally the Taylor approximation $\ln(1 + k/(N-k)) = k/N + O(k^2/N^2)$ yields

$$\begin{aligned} \ln\binom{N}{k} &= k\ln(N/k) + (N-k)[k/N + O(k^2/N^2)] + O(1/N) \\ &= k\ln(N/k) + k(N-k)/N + O(k^2/N) \\ &= k\ln(N/k) + k(1-k/N) + O(k^2/N) \\ &= k\ln(N/k) + k + O(k^2/N) \\ &= k\ln(eN/k) + O(k^2/N). \end{aligned}$$

C Basic Results on Random Vectors

The following is a well-known estimate for the tail of a Gaussian.

Proposition 5. *When Z is a unit Gaussian, we have*

$$\ln \mathrm{P}(Z \geq a) = -\frac{1}{2}a^2 - \ln a - \frac{1}{2}\ln 2\pi + o(1).$$

for $a \to +\infty$.

In fact, the leading-order term $-\frac{1}{2}a^2$ is an upper bound on $\ln \mathrm{P}(Z \geq a)$. One simple way to establish such estimates on tail probabilities is via the Chernoff inequality: for any random variable X and $\lambda > 0$,

$$\mathrm{P}(X \geq a) \leq e^{-\lambda a}\,\mathrm{E}\,e^{\lambda X} = \exp(-\lambda a + K_X(\lambda))$$

where $K_X(\lambda) = \ln \mathrm{E}\,e^{\lambda X}$ is the cumulant generating function. In general, putting $\lambda = a$ in the Chernoff inequality for a random variable X with $K_X(\lambda) \leq \lambda^2/2$ gives

$$\mathrm{P}(X \geq a) \leq \exp(-a^2/2),$$

which is called a sub-Gaussian tail bound.

Let $X_1, \ldots, X_k$ be independent Rademacher random variables, each uniformly distributed on $\{-1, 1\}$. Then $K_{X_i}(\lambda) = \ln\cosh\lambda \leq \lambda^2/2$, so

$$K_{R_k}(\lambda) = \sum_{i=1}^{k} K_{X_i}(\lambda) \leq k\lambda^2/2$$

where $R_k = X_1 + \cdots + X_k$. This proves the following.

Proposition 6 (Chernoff bound for Rademacher sums)**.** *For R_k a sum of k independent Rademacher variables and $t > 0$,*

$$\mathrm{P}(R_k \geq t) \leq \exp\left(-\frac{t^2}{2k}\right).$$

This is enough to prove Proposition 1, which we restate here for convenience.

Proposition. *Let $d > 2\epsilon^{-2}(2\ln N + \ln p^{-1})$, and let*

$$\{F_1, \ldots, F_N\} \subseteq \{-1/\sqrt{d}, 1/\sqrt{d}\}^d$$

be random vectors with independent, uniformly distributed entries. Then $|\langle F_i, F_j\rangle| < \epsilon$ for all $i \neq j$ with probability at least $(1-p)$.

Proof. Each inner product $\langle F_i, F_j\rangle$ is distributed like R_d/d where R_d is a sum of d Rademacher variables. By the Chernoff bound, $\mathrm{P}(\langle F_i, F_j\rangle \geq \epsilon) = \mathrm{P}(R_d \geq d\epsilon) \leq \exp(-d\epsilon^2/2)$. By symmetry and a union bound over all $\binom{N}{2} < N^2/2$ pairs,

$$\mathrm{P}(\exists\, i \neq j : |\langle F_i, F_j\rangle| \geq \epsilon) \leq N^2 \exp(-d\epsilon^2/2).$$

This is at most p when $d \geq 2\epsilon^{-2}(2\ln N + \ln p^{-1})$. □

The interested reader should also compare this result to the Johnson-Lindenstrauss lemma, which is proved in a very similar way. See Dasgupta & Gupta (2003) for a proof, or the last section of Foucart & Rauhut (2013) for a discussion of the JL lemma with some broader context.

Now, consider the inner product of a pair of d-dimensional codewords (X_i, X_j) drawn from a random spherical dictionary. (That is, X_i and X_j are uniformly distributed on the unit sphere in d dimensions.) Write $b(d,t)$ for the probability that $\langle X_i, X_j\rangle \geq t$. This number can also be viewed as fraction of the unit sphere occupied by the spherical cap

$$\{(v_1, \ldots, v_d) : v_1^2 + \cdots + v_d^2 = 1,\ v_1 \geq t\}$$

Using the fact that normalizing a Gaussian random vector gives a uniform draw from the unit sphere, it is possible to establish that

$$b(d,t) \leq (1 + o(1)) \exp\left(-\frac{t^2 d}{2}\right)$$

using Chernoff inequalities. However, for the proof of Proposition 3, we also need a *lower* bound on this tail probability. This is accomplished by the following result, which is a corollary of some ideas from Lecture 2 of Ball (1997).

Proposition 7. *For all $t \in (0,1)$,*

$$\exp\left(-\frac{t^2 d}{2(1-t^2)}\right) \leq b(d,t) \leq \exp\left(-\frac{t^2 d}{2}\right)$$

D Proofs of Main Results

We begin with Proposition 3, restated here for convenience. Recall that $b(d,k,N;\text{top-}k)$ is the probability that the top-k decoder fails to recover a k-sparse subset of $[N]$ from a d-dimensional superposition code.

Proposition. *Let $\epsilon > 0$ be arbitrary. Over the regime*

$$d \leq (1-\epsilon)2k\ln N, \quad 2 \leq k < N/2,$$

for random spherical dictionaries it holds that

$$\lim_{N\to\infty} b(d,k,N;\textit{top-}k) = 1.$$

The proof can likely also be modified to hold for Rademacher dictionaries.

Proof. We can suppose w.l.o.g. that X is the set $\{1, \ldots, k\}$. As before, we write F for the dictionary, F_i for its constituent codewords, and $Y = FX = F_1 + \cdots + F_k$ for the superposition code.

It is easy to show that, with probability bounded below by a constant independent of (d, k), we have $\|Y\| \geq \sqrt{k}$ and $\langle F_1, Y \rangle < 1$ so long as $k > 2$.

Let us condition on this event. The remaining codewords $F_{k+1}, \ldots, F_N$ are independent, uniform draws from the unit sphere. Let B_i be the event that $\langle F_i, Y \rangle \geq 1$, and let B be the union of the $(N-k)$ events $\{B_{k+1}, \ldots, B_N\}$. If B holds, then top-k decoding will not correctly recover X from the code Y.

For each $i \in \{k+1, \ldots, N\}$, we have

$$\mathrm{P}(B_i) = \mathrm{P}(\langle F_i, Y \rangle \geq 1) = \mathrm{P}\left(\left\langle F_i, \frac{Y}{\|Y\|} \right\rangle \geq \frac{1}{\|Y\|} \right).$$

For each $i > k$, the pair $(F_i, Y/\|Y\|)$ is a pair of independent, uniform draws from the unit sphere, so combining Proposition 7 with our assumption $\|Y\| \geq \sqrt{k}$ gives

$$\mathrm{P}(B_i) \geq \exp\left(-\frac{d}{2(\|Y\|^2 - 1)} \right) > \exp\left(-\frac{d}{2(k-1)} \right).$$

Since the events B_i are independent, we can lower bound $\mathrm{P}(B)$ by

$$\mathrm{P}(B) = \mathrm{P}\left(\bigcup_i B_i \right) = 1 - \prod_{i=k+1}^{N} (1 - \mathrm{P}(B_i)) \geq 1 - \exp\left(\sum_i - \mathrm{P}(B_i) \right).$$

Substituting in our lower bound on $\mathrm{P}(B_i)$ gives

$$\mathrm{P}(B) \geq 1 - \exp\left(-(N-k) \cdot \exp\left(-\frac{d}{2(k-1)} \right) \right).$$

When $d \leq (1-\epsilon) 2k \ln N$, we obtain

$$\begin{aligned} \mathrm{P}(B) &\geq 1 - \exp\left(-(N-k) \exp\left(-\frac{(1-\epsilon) k \ln N}{k-1} \right) \right) \\ &\geq 1 - \exp\left(-(N-k) \exp(-(1-\epsilon) \ln N) \right). \end{aligned}$$

Finally, by our assumptions $k < N/2$ and $\epsilon > 0$, we conclude that

$$b(d, k, N; \text{top-}k) \geq \mathrm{P}(B) \geq 1 - \exp\left(-\frac{1}{2} \exp(\epsilon \ln N) \right) \xrightarrow{N \to \infty} 1.$$

□

Now we turn to the proof of Proposition 4, restated below. Recall that $b(d, k, N; \tau)$ is the failure probability for threshold decoding at level τ.

Proposition. *For $\eta \in (0, 1]$, let $k \leq N^\eta$. Then for any $\epsilon > 0$, over the regime*

$$d \geq (1+\epsilon) 2 (1 + \sqrt{\eta})^2 k \ln N,$$

for Rademacher dictionaries it holds that

$$\lim_{N \to \infty} b(d, k, N; \tau(\eta)) = 0$$

where $\tau(\eta) = (1 + \sqrt{\eta})^{-1}$.

Our proof relies on the following bound for $b(d, k, N; \tau)$, valid for Rademacher dictionaries. Similar bounds can be proven for other random dictionaries.

Lemma 1. *For all $d, k, N \in \mathbb{N}$ and $\tau \in (0, 1)$,*

$$b(d, k, N; \tau) \leq k \exp\left(-\frac{(1-\tau)^2 d}{2(k-1)}\right) + (N-k) \exp\left(-\frac{\tau^2 d}{2k}\right).$$

Proof. Suppose w.l.o.g. that $X = \{1, \ldots, k\}$ and let $Y = FX = F_1 + \cdots + F_k$.

Define the families of events

$$\begin{aligned} A_i &= [\langle F_i, Y\rangle < \tau] \quad \text{for } 1 \leq i \leq k, \\ B_i &= [\langle F_i, Y\rangle \geq \tau] \quad \text{for } i > k. \end{aligned}$$

Let R_t be a sum of t independent Rademacher variables, as in Proposition 6. For $i > k$, $\langle F_i, Y\rangle$ is distributed as R_{kd}/d. Thus, by a Chernoff bound,

$$\mathrm{P}(B_i) = \mathrm{P}(\langle F_i, Y\rangle \geq \tau) = \mathrm{P}(R_{kd} \geq d\tau) \leq \exp\left(-\frac{\tau^2 d}{2k}\right).$$

Similarly, for $i \leq k$, $\langle F_i, Y\rangle = \langle F_i, F_i\rangle + \langle F_i, \sum_{j \neq i} F_j\rangle$ is distributed like $1 + R_{(k-1)d}/d$, and so

$$\begin{aligned} \mathrm{P}(A_i) &= \mathrm{P}(\langle F_i, Y\rangle < \tau) = \mathrm{P}(R_{(k-1)d} < (\tau - 1)d) \\ &\leq \mathrm{P}(R_{(k-1)d} \geq (1-\tau)d) \leq \exp\left(-\frac{(1-\tau)^2 d}{2(k-1)}\right). \end{aligned}$$

Our conclusion follows from a union bound

$$b(d, k, N; \tau) = \mathrm{P}\left(\bigcup_i A_i \cup \bigcup_i B_i\right) \leq \sum_{i=1}^{k} \mathrm{P}(A_i) + \sum_{i=k+1}^{N} \mathrm{P}(B_i).$$

□

We are now prepared to complete the proof.

Proof of Proposition 4. Put $d = Ck \ln N$ for some constant C to be determined. Using the assumption $k \leq N^\eta$ and the previous lemma lets us bound $b(d, k, N; \tau)$ by a sum of powers of N with exponents depending only on η and C:

$$\begin{aligned} b(d, k, N; \tau) &\leq N^\eta \exp\left(-\frac{(1-\tau)^2 Ck \ln N}{2k}\right) + N \exp\left(-\frac{\tau^2 Ck \ln N}{2k}\right) \\ &= \exp\left(\left(\eta - \frac{(1-\tau)^2 C}{2}\right) \ln N\right) + \exp\left(\left(1 - \frac{\tau^2 C}{2}\right) \ln N\right). \end{aligned} \tag{7}$$

Putting $\tau = (1 + \sqrt{\eta})^{-1}$,

$$(1-\tau)^2 = \frac{\eta}{(1+\sqrt{\eta})^2} \quad \text{and} \quad \tau^2 = \frac{1}{(1+\sqrt{\eta})^2}$$

and so

$$b(d, k, N; \tau) \leq \exp\left(\left(\eta - \frac{\eta C}{2(1+\sqrt{\eta})^2}\right) \ln N\right) + \exp\left(\left(1 - \frac{C}{2(1+\sqrt{\eta})^2}\right) \ln N\right).$$

As $N \to \infty$, both terms vanish for any fixed $C > 2(1 + \sqrt{\eta})^2$, which completes the proof. □

In the proof above, the choice of threshold $\tau = (1+\sqrt{\eta})^{-1}$ is only justified post hoc. We now explain how to derive this expression and how it relates to the MAP threshold derived in Section 2.

For a given η, we're interested in knowing the critical value of C for which some choice of threshold τ allows both terms in (7) to converge to 0, meaning that both exponents are negative:

$$\min_{\tau\in(0,1)} \max\left\{\eta - \frac{(1-\tau)^2 C}{2}, 1 - \frac{\tau^2 C}{2}\right\} < 0.$$

Let's write $F(\eta,\tau,C)$ for the maximum of the two exponents. The minimizer in τ is $\tau = 1/2 + (1-\eta)/C$, at which point both exponents are equal and

$$F(\eta,\tau,C) = \frac{1+\eta}{2} - \frac{C}{8} - \frac{(1-\eta)^2}{2C}.$$

Solving a quadratic shows that this expression is negative so long as $C > 2(1+\sqrt{\eta})^2$. Substituting this critical value into our expression for the threshold τ achieving the best exponents in the upper bound gives

$$\tau = \frac{1}{2} + \frac{1-\eta}{C} = \frac{1}{2} + \frac{1-\eta}{2(1+\sqrt{\eta})^2} = \frac{1}{2} + \frac{(1-\sqrt{\eta})(1+\sqrt{\eta})}{2(1+\sqrt{\eta})^2} = \frac{1}{2} + \frac{1-\sqrt{\eta}}{2(1+\sqrt{\eta})} = \frac{1}{1+\sqrt{\eta}}.$$

Since $F(\eta,\tau,C)$ is decreasing in C, putting $\tau = (1+\sqrt{\eta})^{-1}$ guarantees that $F(\eta,\tau,C) < 0$ for any C satisfying $\min_\tau F(\eta,\tau,C) < 0$.

Finally, note that the threshold minimizing the maximum exponent F of our Chernoff bounds agrees with the threshold we derived from a maximum a posteriori estimate in Section 2 (Equation (4)). Indeed, substituting $C = d/(k\ln N)$ and $\eta = \ln k/\ln N$ gives

$$\tau_{\text{Chernoff}} = \frac{1}{2} + \frac{1-\eta}{C} = \frac{1}{2} + \frac{k\ln N}{d} - \frac{k\ln k}{d} = \frac{1}{2} - \frac{k}{d}\ln\frac{k}{N}$$

while, following our earlier discussion, the maximum a posteriori threshold is

$$\begin{aligned}\tau_{\text{MAP}} &= \frac{1}{2} - \rho^{-1}\ln\frac{\mathrm{P}(X_i=1)}{\mathrm{P}(X_i=0)} = \frac{1}{2} - \frac{N-1}{N}\frac{k}{d}\left(\ln\frac{k}{N} + \frac{k}{N} + O\left(\frac{k^2}{N^2}\right)\right)\\ &\approx \frac{1}{2} - \frac{k}{d}\ln\frac{k}{N}\end{aligned}$$

in our regime $k \ll N$, $k \ll d$.