Hiding Data Helps: On the Benefits of Masking for Sparse Coding

Muthu Chidambaram [1] **Chenwei Wu** [1] **Yu Cheng** [2] **Rong Ge** [1]

Abstract

Sparse coding, which refers to modeling a signal as sparse linear combinations of the elements of a learned dictionary, has proven to be a successful (and interpretable) approach in applications such as signal processing, computer vision, and medical imaging. While this success has spurred much work on provable guarantees for dictionary recovery when the learned dictionary is the same size as the ground-truth dictionary, work on the setting where the learned dictionary is larger (or *over-realized*) with respect to the ground truth is comparatively nascent. Existing theoretical results in this setting have been constrained to the case of noise-less data. We show in this work that, in the presence of noise, minimizing the standard dictionary learning objective can fail to recover the elements of the ground-truth dictionary in the over-realized regime, regardless of the magnitude of the signal in the data-generating process. Furthermore, drawing from the growing body of work on self-supervised learning, we propose a novel masking objective for which recovering the ground-truth dictionary is in fact optimal as the signal increases for a large class of data-generating processes. We corroborate our theoretical results with experiments across several parameter regimes showing that our proposed objective also enjoys better empirical performance than the standard reconstruction objective.

1. Introduction

Modeling signals as sparse combinations of latent variables has been a fruitful approach in a variety of domains, and has been especially useful in areas such as medical imaging (Zhang et al., 2017), neuroscience (Olshausen & Field, 2004), and genomics (Tibshirani & Wang, 2008), where learning parsimonious representations of data is of high importance. The particular case of modeling data in some high-dimensional space $\mathbb{R}^d$ as sparse *linear* combinations of a set of p vectors in $\mathbb{R}^d$ (referred to as a *dictionary*) has received significant attention over the past two decades, leading to the development of many successful algorithms and theoretical frameworks.

In this case, the typical assumption is that we are given data y_i generated as $y_i \sim Az_i + \epsilon_i$, where $A \in \mathbb{R}^{d\times p}$ is a ground truth dictionary, z_i is a sparse vector, and ϵ_i is some potentially non-zero noise. When the dictionary A is known a priori, the goal of modeling is to recover the sparse representations z_i, and the problem is referred to as *compressed sensing*. However, in many applications we do not have access to the ground truth A, and instead hope to simultaneously learn a dictionary B that approximates A along with learning sparse representations of the data.

This problem is referred to as *sparse coding* or *sparse dictionary learning*, and is the focus of this work. One of the primary goals of analyses of sparse coding is to provide provable guarantees on how well one can hope to recover the ground truth dictionary A, both with respect to specific algorithms and information theoretically. Prior work on such guarantees has focused almost exclusively on the setting where the learned dictionary B also belongs to $\mathbb{R}^{d\times p}$ (same space as the ground truth), which is in line with the fact that recovery error is usually formulated as some form of the Frobenius norm of the difference between B and A.

Unfortunately, in practice, one does not necessarily have access to the structure of A, and it is thus natural to consider what happens (and how to formulate recovery error) when learning a $B \in \mathbb{R}^{d\times p'}$ with $p' \neq p$. Of particular interest is the case where $p' > p$, where it is possible to recover A as a sub-dictionary of B.

The study of this *over-realized* setting was recently taken up in the work of Sulam et al. (2020), in which the authors showed (perhaps surprisingly) that a modest level of over-realization can be empirically and theoretically beneficial.

However, the results of Sulam et al. (2020) are restricted to the noise-less setting where data is generated simply as $y_i \sim Az_i$. We thus ask the following questions:

[1]Department of Computer Science, Duke University [2]Department of Computer Science, Brown University. Correspondence to: Muthu Chidambaram <muthu@cs.duke.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

Does over-realized sparse coding run into pitfalls when there is noise in the data-generating process? And if so, is it possible to prevent this by designing new sparse coding algorithms?

1.1. Main Contributions and Outline

In this work, we answer both of these questions in the affirmative. After providing the necessary background on sparse coding in Section 2, we show in Theorem 3.2 of Section 3 that, as intuition would lead one to suspect, using standard sparse coding algorithms for learning over-realized dictionaries in the presence of noise leads to overfitting. In fact, our result shows that even if we allow the algorithms access to infinitely many samples and allow for solving NP-hard optimization problems, the learned dictionary B can still fail to recover A.

The key idea behind this result is that existing approaches to sparse coding rely largely on a two-step procedure (outlined in Algorithm 1) of solving the compressed sensing problem $B\hat{z} = y_i$ for a learned dictionary B, and then updating B based on a reconstruction objective $\|y_i - B\hat{z}\|^2$. However, because we force $\hat{z}$ to be sparse, by choosing B to have columns that correspond to linear combinations of the columns of A, we can effectively "cheat" and get around the sparsity constraint on $\hat{z}$. In this way, it can be optimal for reconstructing the data y_i to not recover A as a sub-dictionary of B.

On the other hand, we show in Theorem 3.6 that for a large class of data-generating processes, it is possible to prevent this kind of cheating in B by performing the compressed sensing step on a subset of the dimensions y_i and computing the reconstruction loss on the complement of that subset. This is the idea of *masking* that has seen great success in large language modeling (Devlin et al., 2019), and our result shows that it can lead to provable benefits even in the context of sparse coding.

Finally, in Section 4 we conduct experiments comparing the standard sparse coding approach to our masking approach across several parameter regimes. In all of our experiments, we find that the masking approach leads to better ground truth recovery, with this being more pronounced as the amount of over-realization increases.

1.2. Related Work

Compressed Sensing. The seminal works of Candes et al. (2006), Candes & Tao (2006), and Donoho (2006) established conditions on the dictionary $A \in \mathbb{R}^{d \times p}$, even in the case where $p \gg d$ (the *overcomplete* case), under which it is possible to recover (approximately and exactly) the sparse representations z_i from $Az_i + \epsilon_i$. In accordance with these results, several efficient algorithms based on convex programming (Tropp, 2006; Yin et al., 2008), greedy approaches (Tropp & Gilbert, 2007; Donoho et al., 2006; Efron et al., 2004), iterative thresholding (Daubechies et al.; Maleki & Donoho, 2010), and approximate message passing (Donoho et al., 2009; Musa et al., 2018) have been developed for solving the compressed sensing problem. There has also been work on modifying these approaches to include a cross-validation step (Boufounos et al., 2007; Ward, 2009), which is similar to the idea of our masking objective. For comprehensive reviews on the theory and applications of compressed sensing, we refer the reader to the works of Candes & Wakin (2008) and Duarte & Eldar (2011).

Sparse Coding. Different framings of the sparse coding problem exist in the literature (Krause & Cevher, 2010; Bach et al., 2008; Zhou et al., 2009), but the canonical formulation involves solving a non-convex optimization problem. Despite this hurdle, a number of algorithms (Engan et al., 1999; Aharon et al., 2006a; Mairal et al., 2010; Arora et al., 2013; 2014; 2015) have been established to (approximately) solve the sparse coding problem under varying conditions, dating back at least to the groundbreaking work of Olshausen & Field (1997) in computational neuroscience. A summary of convergence results and the conditions required on the data-generating process for several of these algorithms may be found in Table 1 of Gribonval et al. (2014).

In addition to algorithm-specific analyses, there also exists a complementary line of work on characterizing the optimization landscape of dictionary learning. This type of analysis is carried out by Gribonval et al. (2014) in the general setting of an overcomplete dictionary and noisy measurements with possible outliers, extending the previous line of work of Aharon et al. (2006b), Gribonval & Schnass (2010), and Geng et al. (2011).

However, as mentioned earlier, these theoretical results rely on learning dictionaries that are the same size as the ground truth. To the best of our knowledge, the over-realized case has only been studied by Sulam et al. (2020), and our work is the first to analyze over-realized sparse coding in the presence of noise.

Self-Supervised Learning. Training models to predict masked out portions of the input data is an approach to self-supervised learning that has led to strong empirical results in the deep learning literature (Devlin et al., 2019; Yang et al., 2019; Brown et al., 2020; He et al., 2022). This success has spurred several theoretical studies analyzing how and why different self-supervised tasks can be used to improve model training (Tsai et al., 2020; Lee et al., 2021; Tosh et al., 2021). The most closely related works to our own in this regard have studied the use of masking objectives in autoencoders (Cao et al., 2022; Pan et al., 2022) and hidden Markov models (Wei et al., 2021).

2. Preliminaries and Setup

We first introduce some notation that we will use throughout the paper.

Notation. Given $n \in \mathbb{N}$, we use $[n]$ to denote the set $\{1, 2, ..., n\}$. For a vector x, we write $\|x\|$ for the $\mathcal{L}_2$-norm of x and $\|x\|_0$ for the number of non-zeros in x. We say a vector x is k-sparse if $\|x\|_0 \leq k$ and we use $\text{supp}(x)$ to denote the support of x. For a vector $x \in \mathbb{R}^d$ and a set $S \subseteq [d]$, we use $[x]_S \in \mathbb{R}^{|S|}$ to denote the restriction of x to those coordinates in S.

For a matrix A, we use A_i to denote the i-th column of A. We write $\|A\|_F$ for the Frobenius norm of A, and $\|A\|_{op}$ for the operator norm of A, and we write $\sigma_{\min}(A)$ and $\sigma_{\max}(A)$ for the minimum and maximum singular values of A. For a matrix $A \in \mathbb{R}^{d \times q}$ and $S \subseteq [q]$, we use $A_S \in \mathbb{R}^{d \times |S|}$ to refer to A restricted to the columns whose indices are in S. We use I_d to denote the $d \times d$ identity matrix. Finally, for $M \subseteq [d]$, we use $P_M \in \mathbb{R}^{|M| \times d}$ to refer to the matrix whose action on x is $P_M x = [x]_M$. Note that for a $d \times q$ matrix A, $P_M A$ would give a subset of rows of A, which is different from the earlier notation A_S which gives a subset of columns.

2.1. Background on Sparse Coding

We consider the sparse coding problem in which we are given measurements $y \in \mathbb{R}^d$ generated as $Az + \epsilon$, where $A \in \mathbb{R}^{d \times p}$ is a ground-truth dictionary, $z \in \mathbb{R}^p$ is a k-sparse vector distributed according to a probability measure $\mathbb{P}_z$, and $\epsilon \in \mathbb{R}^d$ is a noise term with i.i.d. entries. The goal is to use the measurements y to reconstruct a dictionary B that is as close as possible to the ground-truth dictionary A.

In the case where B has the same dimensions as A, one may want to formulate this notion of "closeness" (or recovery error) as $\|A - B\|_F^2$. However, directly using the Frobenius norm of $(A - B)$ is too limited, as it is sufficient to recover the columns of A up to permutations and sign flips. Therefore, a common choice of recovery error (Gribonval et al., 2014; Arora et al., 2015) is the following:

$$\min_{P \in \Pi} \|A - BP\|_F^2 \tag{2.1}$$

where Π is the set of orthogonal matrices whose entries are 0 or ± 1.

In the over-realized setting, when $B \in \mathbb{R}^{d \times p'}$ with $p' > p$, Equation (2.1) no longer makes sense as A and B do not have the same size. In this case, one can generalize Equation (2.1) to measure the distance between each column of A and the column closest to it in B (up to change of sign). This notion of recovery was studied by Sulam et al. (2020), and we use the same formulation in this work:

$$d_R(A, B) \triangleq \frac{1}{p} \sum_{i=1}^{p} \min_{j \in [p'], c \in \{-1,1\}} \|A_i - cB_j\|^2 \tag{2.2}$$

Note that Equation (2.2) introduced the coefficient $1/p$ in the recovery error and thus corresponds to the *average* distance between A_i and its best approximation in B. Also, Equation (2.2) only allows sign changes, even though for reconstructing Az, it is sufficient to recover the columns of A up to arbitrary scaling. In our experiments we enforce A and B to have unit column norms so a sign change suffices; in theory one can always modify the B matrix to have correct norm so it also does not change our results.

Given access to only measurements y, the algorithm cannot directly minimize the recovery error $d_R(A, \cdot)$. Instead, sparse coding algorithms often seek to minimize the following surrogate loss:

$$\ell(B) = \mathbb{E}_y \left[\min_{\hat{z} \in \mathbb{R}^{p'}} \|y - B\hat{z}\|^2 + h(\hat{z}) \right] \tag{2.3}$$

where h is a sparsity-promoting penalty function. Typical choices of h include hard sparsity ($h(\hat{z}) = 0$ if $\hat{z}$ is k-sparse and $h(\hat{z}) = \infty$ otherwise) as well as the $\mathcal{L}_1$ penalty $h(\hat{z}) = \|\hat{z}\|_1$. While hard sparsity is closer to the assumption on the data-generating process, it is well-known that optimizing under exact sparsity constraints is NP-hard in the general case (Natarajan, 1995). When $h(\hat{z}) = \|\hat{z}\|_1$ is used, the learning problem is also known as basis pursuit denoising (Chen & Donoho, 1994) or Lasso (Tibshirani, 1996).

Equation (2.3) is the population loss one wishes to minimize when learning a dictionary B. In practice, sparse coding algorithms must work with a finite number of measurements $y_1, y_2, \ldots, y_n$ obtained from the data-generating process and instead minimize the empirical loss $\tilde{\ell}(B)$:

$$\tilde{\ell}(B) = \sum_{i=1}^{n} \min_{\hat{z} \in \mathbb{R}^{p'}} \|y_i - B\hat{z}\|^2 + h(\hat{z}) \tag{2.4}$$

2.2. Sparse Coding via Orthogonal Matching Pursuit

Most existing approaches for optimizing Equation (2.4) can be categorized under the general alternating minimization approach described in Algorithm 1. For simplicity we state Algorithm 1 in terms of a single input signal $y \in \mathbb{R}^d$, but in practice the dictionary update in Algorithm 1 is performed after batching over several input signals.

At iteration t, Algorithm 1 performs a decoding/compressed sensing step using the current learned dictionary $B^{(t)}$ and the input data y. As mentioned in Section 1.2, there are several well-studied algorithms for this decoding step. Because we are interested in enforcing a hard-sparsity constraint, we

Algorithm 1 Alternating Minimization Framework

Input: Data $y \in \mathbb{R}^d$, Dictionary $B^{(t)} \in \mathbb{R}^{d \times p'}$
Decoding Step: Solve $B^{(t)}\hat{z} = y$ for k-sparse $\hat{z}$
Update Step: Update $B^{(t)}$ to $B^{(t+1)}$ by performing a gradient step on loss computed using $B^{(t)}\hat{z}$ and y

restrict our attention to algorithms that are guaranteed to produce a k-sparse representation in the decoding step.

We thus focus on Orthogonal Matching Pursuit (OMP) (Mallat & Zhang, 1993; Rubinstein et al., 2008), which is a simple greedy algorithm for the decoding step. The basic procedure of OMP is to iteratively expand a subset $T \subset [p']$ of atoms (until $|T| = k$) by considering the correlation between the unselected atoms in the current dictionary $B^{(t)}$ and the residual $\left(y - B_T^{(t)} \operatorname{argmin}_{\hat{z} \in \mathbb{R}^{|T|}} \left\|y - B_T^{(t)}\hat{z}\right\|^2\right)$ (i.e., the least squares solution using atoms in T). A more precise description of the algorithm can be found in Rubinstein et al. (2008). Moving forward, we will use $g_{\text{OMP}}(y, B, k)$ to denote the k-sparse vector $\hat{z} \in \mathbb{R}^{p'}$ obtained by running the OMP algorithm on an input dictionary B and a measurement y.

2.3. Conditions on the Data-Generating Process

For the data-generating process $y \sim Az + \epsilon$, it is in general impossible to successfully perform the decoding step in Algorithm 1 even with access to the ground-truth dictionary A. As a result, several conditions have been identified in the literature under which it is possible to provide guarantees on the success of decoding the sparse representation z. We recall two of the most common ones (Candes & Tao, 2005).

Definition 2.1. [*Restricted Isometry Property (RIP)*] We say that a matrix $A \in \mathbb{R}^{d \times p}$ satisfies (s, δ_s)-RIP if the following holds for all s-sparse $x \in \mathbb{R}^p$:

$$(1 - \delta_s)\|x\|^2 \leq \|Ax\|^2 \leq (1 + \delta_s)\|x\|^2 \tag{2.5}$$

Definition 2.2. [*μ-Incoherence*] A matrix $A \in R^{d \times p}$ with unit norm columns is μ-incoherent if:

$$|\langle A_i, A_j \rangle| \leq \mu \quad \text{for all } i \neq j \tag{2.6}$$

These two properties are closely related. For example, as a consequence of the Gershgorin circle theorem, (δ_s/s)-incoherent matrices must satisfy (s, δ_s)-RIP.

Given the prominence of RIP and incoherence conditions in the compressed sensing and sparse coding literature, there has been a large body of work investigating families of matrices that satisfy these conditions. We refer the reader to Baraniuk et al. (2008) for an elegant proof that a wide class of random matrices in $\mathbb{R}^{d \times p}$ (i.e. subgaussian) satisfy (k, δ)-RIP with high probability depending on δ, k, p, and d. For an overview of deterministic constructions of such matrices, we refer the reader to Bandeira et al. (2012) and the references therein.

3. Main Results

Having established the necessary background, we now present our main results. Our first result shows that minimizing the population reconstruction loss with a hard-sparsity constraint can lead to learning a dictionary B that is far from the ground truth. We specifically work with the loss defined as:

$$L(B, k) = \mathbb{E}_y \left[\min_{\|\hat{z}\|_0 \leq k} \|y - B\hat{z}\|^2 \right] \tag{3.1}$$

Note that in the definition of $L(B, k)$, we are considering an NP-hard optimization problem (exhaustively searching over all k-sparse supports). We could instead replace this exhaustive optimization with an alternative least-squares-based approach (so long as it is at least as good as performing least squares on a uniformly random choice of k-sparse support), and our proof techniques for Theorem 3.2 would still work. We consider this version only to simplify the presentation.

We now show that, under appropriate settings, there exists a dictionary B whose population loss $L(B, k)$ is smaller than that of A, while $d_R(A, B)$ is bounded away from 0 by a term related to the noise in the data-generating process.

Assumption 3.1. Let $A \in \mathbb{R}^{d \times p}$ be an arbitrary matrix with unit-norm columns satisfying $(2k, \delta)$-RIP for $k = o\left(\frac{d}{\log p}\right)$ and $\delta = o(1)$, and suppose $\sigma^2_{\min}(A) = \Omega(p/d)$. We assume each measurement y is generated as $y \sim Az + \epsilon$, where z is a random vector drawn from an arbitrary probability measure $\mathbb{P}_z$ on k-sparse vectors in $\mathbb{R}^p$, and $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ for some $\sigma > 0$.

Theorem 3.2. *[Overfitting to Reconstruction Loss] Consider the data-generating model in Assumption 3.1 and define $\Lambda(z)$ to be:*

$$\Lambda(z) = \inf\{t \mid \mathbb{P}_z(\|z\| \geq t) \leq 1/d\}. \tag{3.2}$$

Then for $q = \Omega(p^2 \max(d\sigma^2, \Lambda(z)^2)/\sigma^2)$, there exists a $B \in R^{d \times q}$ such that $L(B, k) \leq L(A, k) - \Omega(k\sigma^2)$ and $d_R(A, B) = \Omega(\sigma^2)$.

Proof Sketch. The key idea is to first determine how much the loss can be decreased by expanding from k-sparse combinations of the columns of A to $2k$-sparse combinations, i.e., lower bound the gap between $L(A, k)$ and $L(A, 2k)$. After this, we can construct a dictionary B whose columns form an ϵ-net (with $\epsilon = \sigma^2$) for all 2-sparse combinations of columns of A. Any $2k$-sparse combination of columns in A can then be approximated as a k-sparse combination of columns in B, which is sufficient for proving the theorem.

Remark 3.3. Before we discuss the implications of Theorem 3.2, we first verify that Assumption 3.1 is not vacuous, and in fact applies to many matrices of interest. This follows from a result of Rudelson & Vershynin (2008), which shows that after appropriate rescaling, rectangular matrices with i.i.d. subgaussian entries satisfy the singular value condition in Assumption 3.1. Furthermore, such matrices will also satisfy the RIP condition so long as k is not too large relative to d and p, as per Baraniuk et al. (2008) as discussed in the previous section.

Theorem 3.2 shows that learning an appropriately over-realized dictionary fails to recover the ground truth *independent* of the distribution of z. This means that even if we let the norm of the signal Az in the data-generating process be arbitrarily large, with sufficient over-realization we may still fail to recover the ground-truth dictionary A by minimizing $L(B, k)$.

We also observe that the amount of over-realization necessary in Theorem 3.2 depends on how well $z \sim \mathbb{P}_z$ can be bounded with reasonably high probability. If z is almost surely bounded (as is frequently assumed), we can obtain the following cleaner corollary of Theorem 3.2.

Corollary 3.4. *Consider the same settings as Theorem 3.2 with the added stipulation that $\mathbb{P}_z(\|z\| \leq C) = 1$ for a universal constant C. Then for $q = \Omega(p^2 d)$, there exists a $B \in R^{d \times q}$ such that $d_R(A, B) = \Omega(\sigma^2)$ and $L(B, k) \leq L(A, k) - \Omega(k\sigma^2)$.*

The reason that we can obtain a smaller population loss than the ground truth in Theorem 3.2 is because we can make use of the extra capacity in B to overfit the noise ϵ in the data-generating process. To prevent this, our key idea is to perform the decoding step $B\hat{z} = y$ on a subset of the dimensions of y - which we refer to as the *unmasked* part of y - and then evaluate the loss of B using the complement of that subset (the *masked* part of y). Intuitively, because each coordinate of the noise ϵ is independent, a dictionary B that well-approximates the noise in the unmasked part of y will have no benefit in approximating the noise in the masked part of y.

We can formalize this as the following masking objective:

$$L_{mask}(B, k, M) = \mathbb{E}_y \left[\left\| [y]_{[d]\setminus M} - [B\hat{z}]_{[d]\setminus M} \right\|^2 \right] \quad (3.3)$$

$$\text{where } \hat{z}([y]_M) = g_{\text{OMP}}([y]_M, P_M B, k) \quad (3.4)$$

In defining L_{mask}, we have opted to use g_{OMP} in the inner minimization step, as opposed to the exhaustive argmin in the definition of L. Similar to the discussion earlier, we could have instead used any other approach based on least squares to decode $\hat{z}$ (including the exhaustive approach), so long as we have guarantees on the probability of failing to recover the true code z given access to the ground-truth dictionary A. This choice of using OMP is mostly to tie our theory more closely with our experiments.

Now we present our second main result which shows, in contrast to Theorem 3.2, that optimizing L_{mask} prevents overfitting noise (albeit in a different but closely related setting).

Assumption 3.5. Let $A \in \mathbb{R}^{d \times p}$ be an arbitrary matrix such that there exists an $M \subset [d]$ with $P_M A$ being μ-incoherent with $\mu \leq C/(2k-1)$ for a universal constant $C < 1$. We assume each measurement y is generated as $y \sim Az + \epsilon$, where $[z]_{\text{supp}(z)} \sim \mathcal{N}(0, \sigma_z^2 \mathrm{I}_k)$ with $\text{supp}(z)$ drawn from an arbitrary probability distribution over all size-k subsets of $[d]$, and $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ for some $\sigma > 0$.

Theorem 3.6. *[Benefits of Masking] Consider the data-generating model in Assumption 3.5. For any non-empty mask $M \subset [d]$ such that $P_M A$ satisfies the μ-incoherence condition in the assumption, we have*

$$\lim_{\sigma_z \to \infty} \left(L_{mask}(A, k, M) - \min_B L_{mask}(B, k, M) \right) = 0 \quad (3.5)$$

That is, as the expected norm of the signal Az increases, there exist minimizers B of L_{mask} such that $d_R(A, B) \to 0$.

Proof Sketch. The proof proceeds by expanding out $L_{mask}(B, k, M)$ and using the fact that $[B\hat{z}^*]_{[d]\setminus M}$ is independent of $[\epsilon]_{[d]\setminus M}$ to obtain a quantity that closely resembles the prediction risk considered in analyses of linear regression. From there we show that the Bayes risk is lower bounded by the risk of a regularized least squares solution with access to a support oracle. We then rely on a result of Cai & Wang (2011) to show that $g_{\text{OMP}}([y]_M)$ recovers the support of z with increasing probability as $\sigma_z \to \infty$, and hence its risk converges to the aforementioned prediction risk.

Remark 3.7. As before, so long as the mask M is not too small (i.e. $\Omega(k\text{polylog}(p))$), matrices with i.i.d. subgaussian entries will satisfy the assumptions on A in Assumption 3.5. In particular, the set of ground truth dictionaries satisfying Assumptions 3.1 *and* 3.5 is non-trivial, once again by results in Rudelson & Vershynin (2008).

Comparing Theorem 3.6 to Theorem 3.2, we see that approximate minimizers of L_{mask} can achieve arbitrarily small recovery error, so long as the signal Az is large enough; whereas for L, there always exist minimizers whose recovery error is bounded away from 0. We note that having the expected norm of the signal be large is effectively necessary to hope for recovering the ground truth in our setting, as in the presence of Gaussian noise there is always some non-zero probability that the decoding step can fail. Full proofs of Theorems 3.2 and 3.6 can be found in Section A of the Appendix.

4. Experiments

In this section, we examine whether the separation between the performance of sparse coding with or without masking (demonstrated by Theorems 3.2 and 3.6) manifests in practice. Code for the experiments in this section can be found at: https://github.com/2014mchidamb/masked-sparse-coding-icml.

For our experiments, we need to make a few concessions from the theoretical settings introduced in Sections 2.1 and 2.3. Firstly, we cannot directly optimize the expectations in L and L_{mask} as defined in Equations (3.1) and (3.3), so we instead optimize the corresponding empirical versions defined in the same vein as Equation (2.4). Another issue is that the standard objective L requires solving the optimization problem $\min_{\|\hat{z}\|_0 \leq k} \|y - B\hat{z}\|^2$, which is NP-hard in general. In order to experiment with reasonably large values of $d, p,$ and p' and to be consistent with the decoding step in L_{mask}, we thus approximately solve the aforementioned optimization problem using OMP.

Algorithm 2 Algorithm for Optimizing L

Input: Data $\{y_1, ..., y_T\}$, Dictionary $B^{(0)} \in \mathbb{R}^{d \times p'}$, Learning Rate $\eta \in \mathbb{R}^+$
for $t = 0$ **to** $T - 1$ **do**
  $z \leftarrow g_{\text{OMP}}(y_{t+1}, B^{(t)})$
  $B^{(t+1)} \leftarrow B^{(t)} - \eta \nabla_{B^{(t)}} \left\| y_{t+1} - B^{(t)} z \right\|^2$
  $B^{(t+1)} \leftarrow \text{Proj}_{\mathbb{S}^{d-1}} B^{(t+1)}$
end for

Algorithm 3 Algorithm for Optimizing L_{mask}

Input: Data $\{y_1, ..., y_T\}$, Dictionary $B^{(0)} \in \mathbb{R}^{d \times p'}$, Learning Rate $\eta \in \mathbb{R}^+$, Mask Size $m \in [d]$
for $t = 0$ **to** $T - 1$ **do**
  $M \leftarrow$ Uniformly random subset of size m from $[d]$
  $z \leftarrow g_{\text{OMP}}([y_{t+1}]_M, B^{(t)})$
  $B^{(t+1)} \leftarrow B^{(t)} - \eta \nabla_{B^{(t)}} \left\| [y_{t+1}]_{M^c} - [B^{(t)} z]_{M^c} \right\|^2$
  $B^{(t+1)} \leftarrow \text{Proj}_{\mathbb{S}^{d-1}} B^{(t+1)}$
end for

The approaches for optimizing L and L_{mask} given n samples from the data-generating process are laid out in Algorithms 2 and 3, in which we use $\text{Proj}_{\mathbb{S}^{d-1}} B$ to denote the result of normalizing all of the columns of B. We also use M^c as a shorthand in Algorithm 3 to denote $[d] \setminus M$.

We point out that Algorithm 3 introduces some features that were not present in the theory of the masking objective; namely, in each iteration, we randomly sample a new mask of a pre-fixed size. This is because if we were to run gradient descent using a single, fixed mask M at each iteration, as we don't differentiate through the OMP steps, the gradient with respect to $B^{(t)}$ computed on the error $\left\| [y]_{[d] \setminus M} - [Bz]_{[d] \setminus M} \right\|^2$ would be non-zero only for those rows of B corresponding to the indices $[d] \setminus M$. To avoid this issue, we sample new masks in each iteration so that each entry of B can be updated. There are alternative approaches that can achieve similar results; i.e. deterministically cycling through different masks, but they have similar performance.

While we will analyze the performance of Algorithms 2 and 3 across several different experimental setups over the next few subsections, we describe the following facets shared across all setups. We generate a dataset of $n = 1000$ samples $y_i = Az_i + \epsilon_i$, where $A \in \mathbb{R}^{d \times p}$ is a standard Gaussian ensemble with normalized columns, the z_i have uniformly random k-sparse supports whose entries are i.i.d. $\mathcal{N}(0, 1)$, and the ϵ_i are mean zero Gaussian noise with some fixed variance (which we will vary in our experiments). We also normalize the z_i so as to constrain ourselves to the bounded-norm setting of Corollary 3.4. In addition to the 1000 samples constituting the dataset, we also assume access to a held-out set of p' samples from the data-generating process for initializing the dictionary $B^{(0)} \in \mathbb{R}^{d \times p'}$.

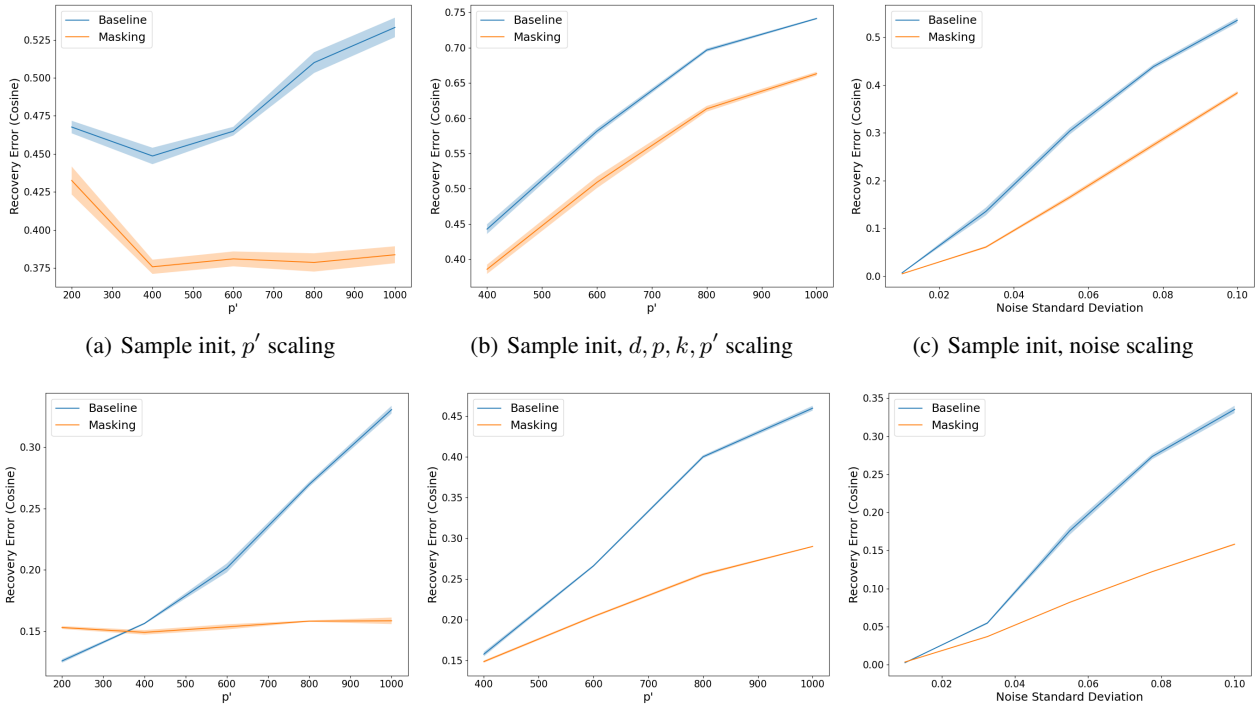
For training, we use batch versions of Algorithms 2 and 3 in which we perform gradient updates with respect to the mean losses computed over $\{y_1, ..., y_B\}$ with $B = 200$ as the batch size. For the actual gradient step, we use Adam (Kingma & Ba, 2014) with its default hyperparameters of $\beta_1 = 0.9, \beta_2 = 0.999$ and a learning rate of $\eta = 0.001$, as we found Adam trains *significantly* faster than SGD (and we ran into problems when using large learning rates for SGD). We train for 500 epochs (passes over the entire dataset) for both Algorithms 2 and 3. For Algorithm 3, we always use a mask size of $d - \lfloor d/10 \rfloor$, which we selected based off of early experiments. We ensured that, even for this fairly large mask size, the gradient norms for both L and L_{mask} were of the same order in our experiments and that 500 epochs were sufficient for training.

We did not perform extensive hyperparameter tuning, but we found that the aforementioned settings performed better than the alternative choices we tested for both algorithms across all experimental setups. Our implementation is in PyTorch (Paszke et al., 2019), and all of our experiments were conducted on a single P100 GPU.

4.1. Scaling Over-realization

We first explore how the choice of p' for the learned dictionary B affects the empirical performance of Algorithms 2 and 3.6 when the other parameters of the problem remain fixed. Theorem 3.2 and Corollary 3.4 indicate that the performance of Algorithm 2 should suffer as we scale p' relative to d and p.

In order to test whether this is actually the case in practice, we consider samples generated as described above with



(a) Sample init, p' scaling (b) Sample init, d, p, k, p' scaling (c) Sample init, noise scaling

(d) Local init, p' scaling (e) Local init, d, p, k, p' scaling (f) Local init, noise scaling

Figure 1. Comparison of Algorithm 2 (Baseline) and Algorithm 3 (Masking) under the settings of Sections 4.1, 4.2, and 4.3 (from left to right). Each curve represents the mean of 5 training runs, with the surrounded shaded area representing one standard deviation.

$A \in \mathbb{R}^{d \times p}$ for $d = 100, p = 200,$ and $\|z\|_0 = k = 5$ fixed, while scaling the number of atoms p' in B from $p' = p$ (exactly realized) to $p' = n$ (over-realized and overparameterized). We choose $\epsilon_i \sim \mathcal{N}(0, 1/d)$, which is a high noise regime as the expected norm of the noise ϵ_i will be comparable to that of the signal Az_i. To make it computationally feasible to run several trials of our experiments, we consider the p' values $\{200, 400, 600, 800, 1000\}$ and do not consider more fine-grained interpolation between p and n.

For the training process, we consider two different initialization of $B^{(0)}$. In the first case, we initialize $B^{(0)}$ to have columns corresponding to the aforementioned set of held-out p' (normalized) samples from the data-generating process, as this is a standard initialization choice that has been known to work well in practice (Arora et al., 2015). However, this initialization choice in effect corresponds to a dataset of $n + p'$ samples, and it is fair to ask whether this initialization benefit is worth the sample cost relative to a random initialization. Our initial experiments showed that this was indeed the case, i.e. random initialization with access to p' additional samples did not perform better, so we focus on this sample-based initialization. That being said, we did not find the ordering of the performances of Algorithms 2 and 3 sensitive to the initializations we considered, only the final absolute performance in terms of $d_R(A, B)$.

In addition to this purely sample-based initialization, we also consider a "local" initialization of $B^{(0)}$ to the ground truth A itself concatenated with $p' - p$ normalized samples. This is obviously not intended to be a practical initialization; the goal here is rather to analyze the extent of overfitting to the noise ϵ_i in the dataset for both algorithms. Namely, we expect that Algorithm 2 will move further away from the ground truth than Algorithm 3.

The results for training using these initializations for both algorithms and then computing the final dictionary recovery errors $d_R(A, B)$ are shown in Figure 1(a, d). We use cosine distance when reporting the error $d_R(A, B)$ since the learned dictionary B also has normalized columns, so Euclidean distance only changes the scale of the error curves and not their shapes.

For both choices of initialization, we observe that Algorithm 3 outperforms Algorithm 2 as p' increases, with this gap only becoming more prominent for larger p'. Furthermore, we find that recovery error actually *worsens* for Algorithm 2 for every choice of $p' > p$ for both initializations in our setting. While this is possibly unsurprising for initializing at the ground truth, it is surprising for the sample-based initialization which does not start at a low recovery error. On the other hand, training using Algorithm 3 improves the

recovery error from initialization when using sample-based initialization for every choice of p' except $p' = n$, which again corresponds to the overparameterized regime in which it is theoretically possible to memorize every sample as an atom of B.

Additionally, we also see that the performance of Algorithm 3 is much less sensitive to the level of over-realization in B. When training from local initialization, Algorithm 3 retains a near-constant level of error/overfitting as we scale p'. Similarly, when training from sample initialization, performance does not degrade as we scale p', and in fact improves initially with a modest level of over-realization.

This improvement up to a certain amount of over-realization (in our case $p' = 2p$) is seen even in the performance of Algorithm 2 for sample initialization (although note that while the recovery error is better for $p' = 2p$ compared to $p' = p$, training still makes the error worse than initialization for Algorithm 2). A similar phenomenon was observed in Sulam et al. (2020) in the setting where $y_i = Az_i$ (no noise), and we find it interesting that the phenomenon is (seemingly) preserved even in the presence of noise. We do not investigate the optimal level of over-realization any further, but believe it would be a fruitful direction for future work.

4.2. Scaling All Parameters

The experiments of Section 4.1 illustrate that for the fixed choices of d, p, and k that we used, scaling the over-realization of B leads to rapid overfitting in the case of Algorithm 2, while Algorithm 3 maintains good performance. To verify that this is not an artifact of the choices of $d, p,$ and k that we made, we also explore what happens when over-realization is kept at a fixed ratio to the other setting parameters while they are scaled.

For these experiments, we consider $A \in \mathbb{R}^{d \times p}$ for $d \in \{100, 150, 200, 250\}$ and scale p as $p = 2d$ and k as $k = \lfloor d/20 \rfloor$ to (approximately) preserve the ratio of atoms and sparsity to dimension from the previous subsection. We choose to scale p' as $p' = 2p$ since that was the best-performing setting (for the baseline) from the experiments of Figure 1. We keep the noise variance at $1/d$ to stay in the relatively high noise regime.

As before, we consider a sample-based initialization as well as a local initialization near the ground truth dictionary A. The results for both Algorithms 2 and 3 under the described parameter scaling are shown in Figure 1(b, e). Once again we find that Algorithm 3 has superior recovery error, with this gap mostly widening as the parameters are scaled. However, unlike the case of fixed $d, p,$ and k, this time the performance of Algorithm 3 also degrades with the scaling. This is to be expected, as increasing p leads to more ground truth atoms that need to be recovered well in order to have small $d_R(A, B)$.

4.3. Analyzing Different Noise Levels

The performance gaps shown in the plots of Figure 1(a, b, d, e) are in the high noise regime, and thus it is fair to ask whether (and to what extent) these gaps are preserved at lower noise settings. We thus revisit the settings of Section 4.1 (choosing $d, p,$ and k to be the same) and fix $p' = 1000$ (the maximum over-realization we consider). We then vary the variance of the noise ϵ_i from $1/d^2$ to $1/d$ linearly, which corresponds to the standard deviations of the noise being $\{0.01, 0.0325, 0.055, 0.0775, 0.1\}$.

Results are shown for the sample-based initialization as well as the local initialization in Figure 1(c, f). Here we see that when the noise variance is very low, there is virtually no difference in performance between Algorithms 2 and 3. Indeed, when the variance is $1/d^2$ we observe that both algorithms are able to near-perfectly recover the ground truth, even from the sample-based initialization.

However, as we scale the noise variance, the gap between the performance of the two algorithms resembles the behavior seen in the experiments of Sections 4.1 and 4.2.

5. Conclusion

In summary, we have shown in Sections 3 and 4 that applying the standard frameworks for sparse coding to the case of learning over-realized dictionaries can lead to overfitting the noise in the data. In contrast, we have also shown that by carefully separating the data used for the decoding and update steps in Algorithm 1 via masking, it is possible to alleviate this overfitting problem both theoretically and practically. Furthermore, the experiments of Section 4.3 demonstrate that these improvements obtained from masking are not at the cost of worse performance in the low noise regime, indicating that a practitioner may possibly use Algorithm 3 as a drop-in replacement for Algorithm 2 when doing sparse coding.

Our results also raise several questions for exploration in future work. Firstly, in both Theorem 3.6 and our experiments we have constrained ourselves to the case of sparse signals that follow Gaussian distributions. It is natural to ask to what extent this is necessary, and whether our results can be extended (both theoretically and empirically) to more general settings (we expect, at the very least, that parts of Assumptions 3.1 and 3.5 can be relaxed). Additionally, we have focused on sparse coding under hard-sparsity constraints and using orthogonal matching pursuit, and it would be interesting to study whether our ideas can be used in other sparse coding settings.

Beyond these immediate considerations, however, the intent of our work has been to show that there is still likely much to be gained from applying ideas from recent developments in areas such as self-supervised learning to problems of a more classical nature such as sparse coding. Our work has only touched on the use of a single such idea (masking), and we hope that future work looks into how other recently popular ideas can potentially improve older algorithms.

Finally, we note that this work has been mostly theoretical in nature, and as such do not anticipate any direct misuses or negative impacts of the results.

Acknowledgements

Rong Ge, Muthu Chidambaram, and Chenwei Wu are supported by NSF Award DMS-2031849, CCF-1845171 (CAREER), CCF-1934964 (Tripods), and a Sloan Research Fellowship. Yu Cheng is supported in part by NSF Award CCF-2307106.

References

Aharon, M., Elad, M., and Bruckstein, A. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006a. doi: 10.1109/TSP.2006.881199.

Aharon, M., Elad, M., and Bruckstein, A. M. On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them. *Linear Algebra and its Applications*, 416(1):48–67, 2006b. ISSN 0024-3795. doi: https://doi.org/10.1016/j.laa.2005.06.035. URL https://www.sciencedirect.com/science/article/pii/S0024379505003459. Special Issue devoted to the Haifa 2005 conference on matrix theory.

Arora, S., Ge, R., and Moitra, A. New algorithms for learning incoherent and overcomplete dictionaries, 2013. URL https://arxiv.org/abs/1308.6273.

Arora, S., Bhaskara, A., Ge, R., and Ma, T. More algorithms for provable dictionary learning. *CoRR*, abs/1401.0579, 2014. URL http://arxiv.org/abs/1401.0579.

Arora, S., Ge, R., Ma, T., and Moitra, A. Simple, efficient, and neural algorithms for sparse coding. *CoRR*, abs/1503.00778, 2015. URL http://arxiv.org/abs/1503.00778.

Bach, F., Mairal, J., and Ponce, J. Convex sparse matrix factorizations, 2008. URL https://arxiv.org/abs/0812.1869.

Bandeira, A. S., Fickus, M., Mixon, D. G., and Wong, P. The road to deterministic matrices with the restricted isometry property, 2012. URL https://arxiv.org/abs/1202.1234.

Baraniuk, R., Davenport, M., DeVore, R., and Wakin, M. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3): 253–263, December 2008. ISSN 0176-4276. doi: 10.1007/s00365-007-9003-x.

Boufounos, P., Duarte, M. F., and Baraniuk, R. G. Sparse signal reconstruction from noisy compressive measurements using cross validation. In *2007 IEEE/SP 14th Workshop on Statistical Signal Processing*, pp. 299–303, 2007. doi: 10.1109/SSP.2007.4301267.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Cai, T. T. and Wang, L. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information Theory*, 57(7):4680–4688, 2011. doi: 10.1109/TIT.2011.2146090.

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

Candes, E., Romberg, J., and Tao, T. 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.

Candes, E. J. and Tao, T. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006. doi: 10.1109/TIT.2006.885507.

Candes, E. J. and Wakin, M. B. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2): 21–30, 2008. doi: 10.1109/MSP.2007.914731.

Cao, S., Xu, P., and Clifton, D. A. How to understand masked autoencoders. *arXiv preprint arXiv:2202.03670*, 2022.

Chen, S. and Donoho, D. Basis pursuit. In *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, volume 1, pp. 41–44. IEEE, 1994.

Daubechies, I., Defrise, M., and De Mol, C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. URL https://arxiv.org/abs/math/0307152.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Donoho, D. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. doi: 10.1109/TIT.2006.871582.

Donoho, D., Elad, M., and Temlyakov, V. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1): 6–18, 2006. doi: 10.1109/TIT.2005.860430.

Donoho, D. L., Maleki, A., and Montanari, A. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, 2009. doi: 10.1073/pnas.0909892106. URL https://www.pnas.org/doi/abs/10.1073/pnas.0909892106.

Duarte, M. F. and Eldar, Y. C. Structured compressed sensing: From theory to applications. *CoRR*, abs/1106.6224, 2011. URL http://arxiv.org/abs/1106.6224.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. Least angle regression. *The Annals of Statistics*, 32(2):407 – 499, 2004. doi: 10.1214/009053604000000067. URL https://doi.org/10.1214/009053604000000067.

Engan, K., Aase, S., and Hakon Husoy, J. Method of optimal directions for frame design. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, volume 5, pp. 2443–2446 vol.5, 1999. doi: 10.1109/ICASSP.1999.760624.

Geng, Q., Wang, H., and Wright, J. On the local correctness of l^1 minimization for dictionary learning. *CoRR*, abs/1101.5672, 2011. URL http://arxiv.org/abs/1101.5672.

Gribonval, R. and Schnass, K. Dictionary identification—sparse matrix-factorization via ℓ_1 -minimization. *IEEE Transactions on Information Theory*, 56(7):3523–3539, 2010. doi: 10.1109/TIT.2010.2048466.

Gribonval, R., Jenatton, R., and Bach, F. R. Sparse and spurious: dictionary learning with noise and outliers. *CoRR*, abs/1407.5155, 2014. URL http://arxiv.org/abs/1407.5155.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16000–16009, June 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

Krause, A. and Cevher, V. Submodular dictionary selection for sparse representation. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pp. 567–574, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

Lee, J. D., Lei, Q., Saunshi, N., and Zhuo, J. Predicting what you already know helps: Provable self-supervised learning. *Advances in Neural Information Processing Systems*, 34:309–323, 2021.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, 11:19–60, mar 2010. ISSN 1532-4435.

Maleki, A. and Donoho, D. L. Optimally tuned iterative reconstruction algorithms for compressed sensing. *IEEE Journal of Selected Topics in Signal Processing*, 4(2): 330–341, 2010. doi: 10.1109/JSTSP.2009.2039176.

Mallat, S. and Zhang, Z. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. doi: 10.1109/78.258082.

Musa, O., Jung, P., and Goertz, N. Generalized approximate message passing for unlimited sampling of sparse signals, 2018. URL https://arxiv.org/abs/1807.03182.

Natarajan, B. K. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995. doi: 10.1137/S0097539792240406.

Olshausen, B. A. and Field, D. J. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311–3325, 1997. ISSN 0042-6989. doi: https://doi.org/10.1016/S0042-6989(97)00169-7. URL https://www.sciencedirect.com/science/article/pii/S0042698997001697.

Olshausen, B. A. and Field, D. J. Sparse coding of sensory inputs. *Current opinion in neurobiology*, 14(4):481–487, 2004.

Pan, J., Zhou, P., and Yan, S. Towards understanding why mask-reconstruction pretraining helps in downstream tasks. *arXiv preprint arXiv:2206.03826*, 2022.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E. Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703, 2019. URL http://arxiv.org/abs/1912.01703.

Rubinstein, R., Zibulevsky, M., and Elad, M. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. 2008.

Rudelson, M. and Vershynin, R. The smallest singular value of a random rectangular matrix. 2008. doi: 10.48550/ARXIV.0802.3956. URL https://arxiv.org/abs/0802.3956.

Sulam, J., You, C., and Zhu, Z. Recovery and generalization in over-realized dictionary learning. *CoRR*, abs/2006.06179, 2020. URL https://arxiv.org/abs/2006.06179.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 00359246. URL http://www.jstor.org/stable/2346178.

Tibshirani, R. and Wang, P. Spatial smoothing and hot spot detection for cgh data using the fused lasso. *Biostatistics*, 9(1):18–29, 2008.

Tosh, C., Krishnamurthy, A., and Hsu, D. Contrastive learning, multi-view redundancy, and linear models. In *Algorithmic Learning Theory*, pp. 1179–1206. PMLR, 2021.

Tropp, J. Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, 2006. doi: 10.1109/TIT.2005.864420.

Tropp, J. A. and Gilbert, A. C. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007. doi: 10.1109/TIT.2007.909108.

Tsai, Y.-H. H., Wu, Y., Salakhutdinov, R., and Morency, L.-P. Self-supervised learning from a multi-view perspective. *arXiv preprint arXiv:2006.05576*, 2020.

Vershynin, R. High-dimensional probability. 2019. URL https://www.math.uci.edu/~rvershyn/papers/HDP-book/HDP-book.pdf.

Ward, R. Compressed sensing with cross validation. *IEEE Transactions on Information Theory*, 55(12):5773–5782, 2009. doi: 10.1109/TIT.2009.2032712.

Wei, C., Xie, S. M., and Ma, T. Why do pretrained language models help in downstream tasks? an analysis of head and prompt tuning. *Advances in Neural Information Processing Systems*, 34:16158–16170, 2021.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

Yin, W., Osher, S., Goldfarb, D., and Darbon, J. Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. *SIAM Journal on Imaging Sciences*, 1(1):143–168, 2008. doi: 10.1137/070703983.

Zhang, R., Shen, J., Wei, F., Li, X., and Sangaiah, A. K. Medical image classification based on multi-scale non-negative sparse coding. *Artificial intelligence in medicine*, 83:44–51, 2017.

Zhou, M., Chen, H., Ren, L., Sapiro, G., Carin, L., and Paisley, J. Non-parametric bayesian dictionary learning for sparse image representations. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper/2009/file/cfecdb276f634854f3ef915e2e980c31-Paper.pdf.

A. Full Proofs

A.1. Proof of Theorem 3.2

We first recall the setting of Theorem 3.2.

Assumption 3.1. Let $A \in \mathbb{R}^{d\times p}$ be an arbitrary matrix with unit-norm columns satisfying $(2k, \delta)$-RIP for $k = o\left(\frac{d}{\log p}\right)$ and $\delta = o(1)$, and suppose $\sigma^2_{\min}(A) = \Omega(p/d)$. We assume each measurement y is generated as $y \sim Az + \epsilon$, where z is a random vector drawn from an arbitrary probability measure $\mathbb{P}_z$ on k-sparse vectors in $\mathbb{R}^p$, and $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ for some $\sigma > 0$.

And now we prove:

Theorem 3.2. *[Overfitting to Reconstruction Loss] Consider the data-generating model in Assumption 3.1 and define $\Lambda(z)$ to be:*

$$\Lambda(z) = \inf\{t \mid \mathbb{P}_z(\|z\| \geq t) \leq 1/d\}. \tag{3.2}$$

Then for $q = \Omega(p^2 \max(d\sigma^2, \Lambda(z)^2)/\sigma^2)$, there exists a $B \in R^{d\times q}$ such that $L(B, k) \leq L(A, k) - \Omega(k\sigma^2)$ and $d_R(A, B) = \Omega(\sigma^2)$.

Proof. Our proof technique will be to first lower bound the gap $L(A, k) - L(A, 2k)$, and then to construct a B matrix that closely approximates the $2k$-sparse combinations of the columns of A.

From the definition of $L(B, k)$ we have that:

$$L(A, k) - L(A, 2k) = \mathbb{E}_y\left[\min_{\|\hat{z}\|_0=k} \|y - A\hat{z}\|^2\right] - \mathbb{E}_y\left[\min_{\|\hat{z}\|_0=2k} \|y - A\hat{z}\|^2\right] \tag{A.1}$$

Now let $\hat{z}^*(y) = \operatorname{argmin}_{\|\hat{z}\|_0=k} \|y - A\hat{z}\|$ and $S^* = \operatorname{supp}(\hat{z}^*)$, and further define:

$$\tilde{z}(y) = \operatorname*{argmin}_{\|\hat{z}\|_0=k} \|(y - A\hat{z}^*(y)) - A\hat{z}\| \tag{A.2}$$

We will also use $\tilde{S} = \operatorname{supp}(\tilde{z}(y))$. For convenience, we will write $\hat{z}^*$ and $\tilde{z}$ when y is clear from context. Applying this notation to Equation (A.1) gives:

$$\begin{aligned} L(A, k) - L(A, 2k) &\geq \mathbb{E}_y[\|y - A\hat{z}^*\|^2] - \mathbb{E}_y[\|y - A\hat{z}^* - A\tilde{z}\|^2] \\ &= \mathbb{E}_y[2\langle y - A\hat{z}^*, A\tilde{z}\rangle] - \mathbb{E}_y[\|A\tilde{z}\|^2] \\ &= \mathbb{E}_y[\|A\tilde{z}\|^2] \end{aligned} \tag{A.3}$$

Where we obtained the last line above by using the fact that $A\tilde{z}$ is the orthogonal projection of $y - A\hat{z}^*$ on to the span of $A_{\tilde{S}}$. Now using the fact that A is $(2k, \delta)$-RIP we have that:

$$\begin{aligned} \mathbb{E}_y[\|A\tilde{z}\|^2] &\geq (1-\delta)^2 \mathbb{E}_y\left[\|\tilde{z}\|^2\right] \\ &\geq (1-\delta)^2 \mathbb{E}_y\left[\left\|A^+_{\tilde{S}}(y - A\hat{z}^*)\right\|^2\right] \\ &\geq (1-\delta)^4 \mathbb{E}_y\left[\left\|A^T_{\tilde{S}}(y - A\hat{z}^*)\right\|^2\right] \\ &= (1-o(1)) \mathbb{E}_y\left[\left\|A^T_{\tilde{S}}(y - A\hat{z}^*)\right\|^2\right] \end{aligned} \tag{A.4}$$

Where above we used the fact that $A^+_{\tilde{S}} = (A^T_{\tilde{S}} A_{\tilde{S}})^{-1} A^T_{\tilde{S}}$ and RIP to obtain $\left\|(A^T_{\tilde{S}} A_{\tilde{S}})^{-1}\right\|_{op} \geq 1/(1+\delta)$, which led to the penultimate step. It remains to compute (or lower bound) the expectation in Equation (A.4). Towards this end, we let S

denote a (uniformly) random subset of size k from $[p]$. Then we have that (using Assumption 3.1):

$$\begin{aligned}\mathbb{E}_y\left[\left\|A_{\tilde{S}}^T(y-A\hat{z}^*)\right\|^2\right] &\geq \mathbb{E}_y\left[\mathbb{E}_S\left[\left\|A_S^T(y-A\hat{z}^*)\right\|^2\right]\right] \\ &= \frac{k}{p}\mathbb{E}_y\left[\left\|A^T(y-A\hat{z}^*)\right\|^2\right] \\ &\geq \frac{k}{p}\sigma_{\min}^2(A)\mathbb{E}_y\left[\|y-A\hat{z}^*\|^2\right] \\ &= \Omega\left(\frac{k}{d}\mathbb{E}_y\left[\|y-A\hat{z}^*\|^2\right]\right) \end{aligned} \tag{A.5}$$

Now the expectation in Equation (A.5) can be lower bounded in the same vein as Equation (A.4) (i.e. relying on the RIP property). Below we use S_{2k}^* to denote the optimal support for the minimization problem.

$$\begin{aligned}\mathbb{E}_y\left[\|y-A\hat{z}^*\|^2\right] &= \mathbb{E}_{z,\epsilon}\left[\min_{\|\hat{z}\|_0=k}\|\epsilon-A(\hat{z}-z)\|^2\right] \\ &\geq \mathbb{E}_\epsilon\left[\min_{\|\hat{z}\|_0=2k}\|\epsilon-A\hat{z}\|^2\right] \\ &= \mathbb{E}_\epsilon\left[\|\epsilon\|^2\right] - 2\mathbb{E}_\epsilon\left[\left\langle A_{S_{2k}^*}^+\epsilon,\epsilon\right\rangle\right] + \mathbb{E}_\epsilon\left[\left\|A_{S_{2k}^*}^+\epsilon\right\|^2\right] \\ &\geq \mathbb{E}_\epsilon\left[\|\epsilon\|^2\right] - (1+o(1))\mathbb{E}_\epsilon\left[\left\|A_{S_{2k}^*}^T\epsilon\right\|^2\right] \\ &\geq d\sigma^2 - (1+o(1))2k\mathbb{E}_\epsilon\left[\max_{i\in[p]}(A_i^T\epsilon)^2\right] \\ &\geq d\sigma^2 - O\left(k\sigma^2\log p\right) \\ &= \Omega(d\sigma^2) \end{aligned} \tag{A.6}$$

Where above we used Lemma A.2 since the random variables $(A_i^T\epsilon)^2$ follows a scaled chi-square distribution with degree of freedom 1, and for the last line we use $k = o\left(\frac{d}{\log p}\right)$. Now putting Equations (A.3)-(A.6) together, we obtain:

$$L(A,k) - L(A,2k) \geq \Omega(k\sigma^2) \tag{A.7}$$

Given the gap between $L(A,k)$ and $L(A,2k)$ shown in Equation (A.7), our goal is now to construct a matrix B such that we can approximate sufficiently large $2k$-sparse combinations of the columns of A via $B\hat{z}$ (where $\hat{z}$ is k-sparse). We recall from standard concentration of measure arguments (see Vershynin (2019) for details) that $\mathbb{P}(\|\epsilon\|^2 \geq 2d\sigma^2) \leq \exp(-\Omega(d))$. Furthermore, by Assumption 3.1, $\|Az\| \leq \Lambda(z)(1+o(1))$ with probability at least $1-1/d$. Thus, we only need the columns of B to approximate Ax for 2-sparse x (since we are interested in $B\hat{z}$ and $\hat{z}$ is k-sparse) and $\|Ax\| \leq \gamma_1\max(\sigma\sqrt{d},\Lambda(z))$ for an appropriately large constant γ_1 (as this will imply we get the same gap as Equation A.6).

To do this, we can construct ϵ-nets for each of the following sets (indexed by the different possible 2-sparse supports $S \subset [p]$):

$$V_S = \{Ax \mid \text{supp}(x) = S,\ \|Ax\| \leq \gamma_1\max(\sigma\sqrt{d},\Lambda(z))\} \tag{A.8}$$

Since A has p columns, we need $\Theta(p^2)$ such ϵ-nets. As long as we choose $\epsilon = \gamma_2\sigma^2$ with γ_2 a constant, we can approximate $2k$-sparse combinations of the columns of A with error $k\gamma_2\sigma^2$ using k-sparse combinations from these nets, which is sufficient for our purposes given the result of Equation (A.7).

Now let the columns of B be the union of the ϵ-nets for the sets V_S and define $\mathcal{E} = \{\|Az\| + \|\epsilon\| \leq \gamma_1\max(\sigma\sqrt{d},\Lambda(z))\}$. After choosing γ_2 to be sufficiently small, we then get from Equations (A.3)-(A.7) and the fact that $\mathbb{P}(\mathcal{E}) \geq 1-1/d$:

$$\begin{aligned}L(A,k) - L(B,k) &\geq \mathbb{E}_y\left[\|A\tilde{z}\|^2 \mid \mathcal{E}\right]\ \mathbb{P}(\mathcal{E}) - k\gamma_2\sigma^2 \\ &= \Omega(k\sigma^2) \end{aligned} \tag{A.9}$$

Noting that the ϵ-nets for each V_S are of size $O(\max(d\sigma^2, \Lambda(z)^2)/\sigma^2)$ from our choice of ϵ (once again, refer to Vershynin (2019) for bounds on the size of ϵ-nets), this construction of B requires $O(p^2 \max(d\sigma^2, \Lambda(z)^2)/\sigma^2)$ columns. As we can choose these columns to be different from those of A by $\gamma_2\sigma^2$ (in norm), we obtain the desired result. □

A.2. Proof of Theorem 3.6

Again, we first recall the setting of Theorem 3.6.

Assumption 3.5. Let $A \in \mathbb{R}^{d\times p}$ be an arbitrary matrix such that there exists an $M \subset [d]$ with $P_M A$ being μ-incoherent with $\mu \leq C/(2k-1)$ for a universal constant $C < 1$. We assume each measurement y is generated as $y \sim Az + \epsilon$, where $[z]_{\text{supp}(z)} \sim \mathcal{N}(0, \sigma_z^2 \mathrm{I}_k)$ with $\text{supp}(z)$ drawn from an arbitrary probability distribution over all size-k subsets of $[d]$, and $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ for some $\sigma > 0$.

In order to prove Theorem 3.6, we will need a result from Cai & Wang (2011), which we restate below.

Theorem A.1 (Theorem 9 in Cai & Wang (2011))**.** *For $y \sim Az + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I})$ and $A \in \mathbb{R}^{d\times p}$ being μ-incoherent with $\mu < 1/(2k-1)$, let us define:*

$$\mathcal{S} = \left\{ A_i : i \in [p],\ |z_i| \geq \frac{2\sigma\sqrt{k}\sqrt{2(1+\eta)\log p}}{1-(2k-1)\mu} \right\} \tag{A.10}$$

Then OMP (as defined in Algorithm) selects a column from $\mathcal{S}$ at each step with probability at least $1 - \frac{1}{p^\eta \sqrt{2\log p}}$ for $\eta \geq 0$.

Now we may prove:

Theorem 3.6. *[Benefits of Masking] Consider the data-generating model in Assumption 3.5. For any non-empty mask $M \subset [d]$ such that $P_M A$ satisfies the μ-incoherence condition in the assumption, we have*

$$\lim_{\sigma_z \to \infty} \left(L_{mask}(A, k, M) - \min_B L_{mask}(B, k, M) \right) = 0 \tag{3.5}$$

That is, as the expected norm of the signal Az increases, there exist minimizers B of L_{mask} such that $d_R(A, B) \to 0$.

Proof. We have from the definition of L_{mask} that:

$$\begin{aligned} L_{mask}(B,k,M) &= \mathbb{E}_{z,\epsilon}\left[\left\| [Az]_{[d]\setminus M} + [\epsilon]_{[d]\setminus M} - [B\hat{z}]_{[d]\setminus M} \right\|^2 \right] \\ &= \mathbb{E}_{z,\epsilon}\left[\left\| [Az]_{[d]\setminus M} - [B\hat{z}]_{[d]\setminus M} \right\|^2 \right] + E_\epsilon\left[\left\| [\epsilon]_{[d]\setminus M} \right\|^2 \right] - \mathbb{E}_{z,\epsilon}\left[\left\langle [B\hat{z}]_{[d]\setminus M}, [\epsilon]_{[d]\setminus M} \right\rangle \right] \\ &= \mathbb{E}_{z,\epsilon}\left[\left\| [Az]_{[d]\setminus M} - [B\hat{z}]_{[d]\setminus M} \right\|^2 \right] + (d - |M|)\sigma^2 \\ &= \mathbb{E}_{z,\epsilon}\left[\left\| P_{[d]\setminus M} Az - P_{[d]\setminus M} B\hat{z} \right\|^2 \right] + (d - |M|)\sigma^2 \end{aligned} \tag{A.11}$$

Since $[B\hat{z}]_{[d]\setminus M}$ is necessarily independent[1] of $[\epsilon]_{d\setminus M}$ (by the construction of $\hat{z}$). Now the quantity in Equation A.11 depending on B looks almost identical to the prediction risk considered in linear regression.

With this in mind, let us define:

$$\mathcal{R}_M(\hat{y}) = \mathbb{E}_{z,\epsilon}\left[\left\| P_{[d]\setminus M} Az - \hat{y} \right\|^2 \right] \tag{A.12}$$

Where $\hat{y}$ is any estimator that depends only on $[y]_M$ (i.e. in the interest of brevity we are omitting writing $\hat{y}([Az+\epsilon]_M)$). We can lower bound Equation (A.11) by analyzing $\mathcal{R}_M(\hat{y})$:

$$\begin{aligned} \inf_{\hat{y}} \mathcal{R}_M &= \inf_{\hat{y}} \mathbb{E}_{z,\epsilon}\left[\left\| P_{[d]\setminus M} Az - \hat{y} \right\|^2 \right] \\ &= \inf_{\hat{y}} \mathbb{E}_{z,\epsilon}\left[\mathbb{E}_{z,\epsilon}\left[\left\| P_{[d]\setminus M} A_{\text{supp}(z)} [z]_{\text{supp}(z)} - \hat{y} \right\|^2 \,\middle|\, \text{supp}(z) \right] \right] \end{aligned} \tag{A.13}$$

[1] There is actually a slight technicality here; we need g_{OMP} to be Borel measurable, which is the case because it consists of the composition of Borel measurable functions.

Equation (A.13) can be lower bounded by considering the infimum over the inner expectation with respect to estimators $\hat{y}$ that have access to $[Az+\epsilon]_M$ and the support $S^* = \text{supp}(z)$. In this case, the Bayes estimator $\hat{y}^*$ is:

$$\begin{aligned}\hat{y}^* &= \mathbb{E}_{z,\epsilon}\left[P_{[d]\setminus M}A_{S^*}[z]_{S^*} \;\middle|\; P_M A_{S^*}[z]_{S^*} + [\epsilon]_M\right] \\ &= P_{[d]\setminus M}A_{S^*}\mathbb{E}_{z,\epsilon}\left[[z]_{S^*} \;\middle|\; P_M A_{S^*}[z]_{S^*} + [\epsilon]_M\right]\end{aligned} \tag{A.14}$$

Since $[z]_{S^*} \sim \mathcal{N}(0, \sigma_z^2 \mathrm{I}_k)$, we can explicitly compute the conditional expectation in Equation (A.14). Indeed, it is just the ridge regression estimator:

$$\hat{y}^* = P_{[d]\setminus M}A_{S^*}\left(\Lambda_{S^*}^T\Lambda_{S^*} + \frac{1}{\sigma_z^2}\mathrm{I}_k\right)^{-1}\Lambda_{S^*}^T(P_M A_{S^*}[z]_{S^*} + [\epsilon]_M) \tag{A.15}$$

Where we have set $\Lambda_{S^*} = P_M A_{S^*}$ above to keep notation manageable. Thus, putting all of the above together we have:

$$L_{mask}(B,k,M) \geq \mathcal{R}_M(\hat{y}^*) + (d - |M|)\sigma^2 \tag{A.16}$$

Now let $\hat{y}_{LS}$ be the least squares estimator with access to the support $\text{supp}(z)$:

$$\hat{y}_{LS} = P_{[d]\setminus M}A_{S^*}\Lambda_{S^*}^+(P_M A_{S^*}[z]_{S^*} + [\epsilon]_M) \tag{A.17}$$

Then we have $\mathcal{R}_M(\hat{y}_{LS}) \to \mathcal{R}_M(\hat{y}^*)$ as $\sigma_z^2 \to \infty$. If we can now show that $\mathcal{R}_M(P_{[d]\setminus M}A\hat{z}) \to \mathcal{R}_M(\hat{y}_{LS})$, then we will be done by Equation (A.16).

Showing this essentially boils down to controlling the error of $\hat{z}$ when OMP fails to recover the true support S^* (because when it recovers the true support, $P_{[d]\setminus M}A\hat{z}$ is exactly $\hat{y}_{LS}$). We do this by appealing to Theorem A.1.

Recall that $[\hat{z}]_S = \Lambda_S^+[y]_M$, where $\Lambda_S = P_M A_S$ with $S = \text{supp}(\hat{z})$ being the support predicted by OMP. Letting $\hat{z}_{LS}$ be the vector whose non-zero components correspond to $[\hat{z}_{LS}]_{S^*} = \Lambda_{S^*}^+[y]_M$, it will suffice to show $\|\hat{z}_{LS} - \hat{z}\|^2 \to 0$ as $\sigma_z \to \infty$, since then we will be done due to the fact that $\left\|P_{[d]\setminus M}A\right\|_{op}$ is constant with respect to σ_z.

Now letting $\tilde{z} = \hat{z} - \Lambda_S^+[\epsilon]_M$ (i.e. $\tilde{z}$ represents the part of the signal z recovered by OMP), we have:

$$\begin{aligned}\|\hat{z}_{LS} - \hat{z}\|^2 &= \left\|z - \tilde{z} + (\Lambda_{S^*}^+ - \Lambda_S^+)[\epsilon]_M\right\|^2 \\ &\leq \|z - \tilde{z}\|^2 + \left\|(\Lambda_{S^*}^+ - \Lambda_S^+)[\epsilon]_M\right\|^2 + 2\|z - \tilde{z}\|\left\|(\Lambda_{S^*}^+ - \Lambda_S^+)[\epsilon]_M\right\|\end{aligned} \tag{A.18}$$

We begin by first analyzing $\|z - \tilde{z}\|^2$. To do so, we introduce the notation $[z]_{U,0}$ to represent the vector in $\mathbb{R}^k$ whose non-zero entries correspond to $[z]_U$ for $U \subset [d]$, $|U| \leq k$. Then we can make use of the following decomposition of $\Lambda_{S^*}[z]_{S^*}$:

$$\Lambda_{S^*}[z]_{S^*} = \Lambda_S[z]_{S^*\cap S,0} + \Lambda_{S^*}[z]_{S^*\setminus S,0} \tag{A.19}$$

From Equation (A.19) we get:

$$\begin{aligned}\|z - \tilde{z}\|^2 &= \left\|[z]_{S^*\setminus S,0} - \Lambda_S^+\Lambda_{S^*}[z]_{S^*\setminus S,0}\right\|^2 \\ &\leq \left\|[z]_{S^*\setminus S,0}\right\|^2 + \left\|(\Lambda_S^T\Lambda_S)^{-1}\Lambda_S^T\Lambda_{S^*}[z]_{S^*\setminus S,0}\right\|^2 + 2\max\left(\left\|[z]_{S^*\setminus S,0}\right\|^2, \left\|\Lambda_S^+\Lambda_{S^*}[z]_{S^*\setminus S,0}\right\|^2\right) \\ &= \sum_{i\in S^*\setminus S} O(z_i^2)\end{aligned} \tag{A.20}$$

where we passed from the penultimate to the last line by using the μ-incoherence of $P_M A$ to control the middle term in the bound. With Equation (A.20) in hand, we are finally in a position to apply Theorem A.1. Let $\eta = C' \log \sigma_z$ for a sufficiently large constant C'. Now for convenience we define:

$$\gamma = \frac{2\sigma\sqrt{k}\sqrt{2(1+\eta)\log p}}{1 - (2k-1)\mu} \tag{A.21}$$

which corresponds to the lower bound in Equation (A.10). Using Theorem A.1 with Equation (A.20) we obtain:

$$\begin{aligned}\mathbb{E}_{z,\epsilon}\left[\|z-\tilde{z}\|^2\right] &\leq \sum_{i\in S^*}\mathbb{P}(\{|z_i|\geq\gamma\}\cap\{i\notin S\})O(\mathbb{E}[z_i^2])+\mathbb{P}(|z_i|<\gamma)O(\gamma^2)\\ &= \sum_{i\in S^*}O\left(\frac{\sigma_z^2}{\sigma_z^{C'}\sqrt{\log p}}+\frac{\gamma^3}{\sigma_z}\right)\end{aligned} \tag{A.22}$$

And clearly Equation (A.22) goes to 0 as $\sigma_z \to \infty$. We can apply similar analysis techniques to the term $\left\|(\Lambda_{S^*}^+ - \Lambda_S^+)[\epsilon]_M\right\|^2$ in Equation (A.18) as well, but for this term we can afford to be less precise.

Namely, when $S = S^*$, this term is 0. The probability that $S \neq S^*$ can be bounded as:

$$\begin{aligned}\mathbb{P}(S\neq S^*) &\leq O\left(\frac{k}{\sigma_z^{C'}\sqrt{\log p}}\right)\mathbb{P}(\min_{i\in S^*}|z_i|\geq\gamma)\\ &= O\left(\frac{k\gamma^k}{\sigma_z^{C'+k}\sqrt{\log p}}\right)\end{aligned} \tag{A.23}$$

where again above we used the naive bound for $\mathbb{P}(\min_{i\in S^*}|z_i|\geq\gamma)$ (i.e. replacing the density with 1 and integrating from $\pm\gamma$). Now we have:

$$\begin{aligned}\mathbb{E}_{z,\epsilon}\left[\left\|(\Lambda_{S^*}^+ - \Lambda_S^+)[\epsilon]_M\right\|^2\right] &\leq \mathbb{P}(S\neq S^*)\mathbb{E}_{z,\epsilon}\left[\left\|\Lambda_{S^*}^+[\epsilon]_M\right\|^2+\left\|\Lambda_S^+[\epsilon]_M\right\|^2+2\max\left(\left\|\Lambda_{S^*}^+[\epsilon]_M\right\|^2,\left\|\Lambda_S^+[\epsilon]_M\right\|^2\right)\right]\\ &\leq \mathbb{P}(S\neq S^*)O\left(k\mathbb{E}_\epsilon\left[\max_{i\in[p]}(A_i^T\epsilon)^2\right]\right)\\ &\leq \mathbb{P}(S\neq S^*)O(k\sigma^2\log p)\end{aligned} \tag{A.24}$$

Putting together Equations (A.22) and (A.24) shows that Equation (A.18) goes to 0 as $\sigma_z \to \infty$, which proves the result.

□

A.3. Auxiliary Lemmas

Lemma A.2. *Let $X_1, \cdots, X_n$ be n chi-square random variables with 1 degree of freedom, then*

$$\mathbb{E}\left[\max_{i\in[n]}X_i\right] = O(\log n).$$

Proof. We bound the maximum via the moment-generating function.

From Jensen's inequality, for $t \in (0, \frac{1}{2})$, we have

$$\exp\left(t\mathbb{E}\left[\max_{i\in[n]}X_i\right]\right) \leq \mathbb{E}\left[\exp\left(t\max_{i\in[n]}X_i\right)\right] \leq \sum_{i=1}^n\mathbb{E}\left[e^{tX_i}\right] = n(1-2t)^{-\frac{1}{2}}.$$

Setting $t = \frac{1}{3}$ gives us

$$\mathbb{E}\left[\max_{i\in[n]}X_i\right] \leq 3\log n - \frac{3}{2}\log\frac{1}{3} = O(\log n).$$

□