Encoded Distributed Optimization

Can Karakus
UCLA, Los Angeles, CA
karakus@ucla.edu

Yifan Sun
Technicolor Research, Los Altos, CA
Yifan.Sun@technicolor.com

Suhas Diggavi
UCLA, Los Angeles, CA
suhasdiggavi@ucla.edu

***Abstract*—Today, many real-world machine learning and data analytics problems are of a scale that requires distributed optimization; unlike in centralized computing, these systems are vulnerable to network and node failures. Recently, coding-theoretic ideas have been applied to mitigate node failures in such distributed computing networks. Relaxing the exact recovery requirement of such techniques, we propose a novel approach for adding redundancy in large-scale convex optimization problems, making solvers more robust against sudden and persistent node failures and loss of data. This is done by linearly encoding the data variables; all other aspects the computation operate as usual. We show that under moderate amounts of redundancy, it is possible to recover a close approximation to the solution under node failures. In particular, we show that encoding with (equiangular) tight frames result in bounded objective error, and obtain an explicit error bound for a specific construction that uses Paley graphs. We also demonstrate the performance of the proposed technique for three specific machine learning problems, (two using real world datasets) namely ridge regression, binary support vector machine, and low-rank approximation.**

I. Introduction

Recent years have seen an enormous surge in interest for large-scale data analytics and machine learning. Typically, solving such large problems require storing data over a large number of distributed nodes and running optimization algorithms over these nodes. In such networks, an important concern is the sudden onset of unresponsive or failed nodes [1]. This can be caused by network failures, background processes, or (in the case of low-cost cloud computing) sudden deallocation of compute resources. In the case of short-term, or intermittent unavailability, such failures can significantly slow down the computation, since speed may be dictated by the slowest node. In longer-term unavailability, it might affect the accuracy of the final solution itself, since a fraction of data is effectively eliminated from the optimization process.

A natural approach to combat node failure is to use redundancy in the form of additional nodes, for example, by simply replicating the data across multiple nodes. However, recently, distributed *coded* computing has received some attention from the information theory community [2], [3], [4], [5]. In particular, [3] used coding-theoretic ideas to provide robustness in two specific linear operations: distributed matrix multiplication and data shuffling. The work in [5] also focused on linear operations, where the idea is to break up large dot products into shorter dot products, and perform redundant copies of the short dot products to provide resilience against failures. On the other hand, [4] considers synchronous gradient descent, and proposes an architecture where each data sample is replicated s times across nodes, and designs a code such that the exact gradient can be recovered as long as fewer than s nodes fail.

In contrast to these works, which mainly focus on adding redundancy in the *implementation* of a distributed algorithm, we embed the redundancy in the *formulation* of the optimization problem. The idea is to linearly encode the data variables in the optimization, place the encoded data in the nodes, and let the nodes operate as if they are solving the original problem, ignoring failed nodes and stragglers. This is inspired by the randomized sketching techniques [6] used for dimensionality reduction in optimization; however, the purpose, operating regime, and the tools used are different in our problem. The main observation underlying our approach is that one needs much less redundancy than in [4] if one backs off from requiring exact recovery of the solution. For instance, for e node failures, the results in [4] imply that one needs a redundancy factor of $e+1$ for exact recovery, whereas we show that the solution can be reasonably approximated with a redundancy factor of 2. Such relaxation is motivated by fields like machine learning, where approximate solutions that achieve good generalization error are sufficient. The main design objective then becomes how to design codes so that with increasing number of failed nodes, the solution accuracy degrades as slowly as possible. In particular, we observe (numerically and analytically) that equiangular tight frames (ETF) are attractive options as coding vectors, since (i) they contain inherent redundancy; (ii), the individual elements provide as much independent information as possible; and (iii), they allow reconstruction of the exact solution when no nodes fail. We also consider random codes, which asymptotically (data length) achieve good performance; however, as numerical evidence suggests, cannot achieve (iii) for finite lengths.

Our approach is not limited to a specific computational operation, but is applicable directly to large class of practically relevant optimization problems; specifically, any optimization that can be formulated as a least-squares minimization over a convex set, including linear regression, support vector machines, compressed sensing, projection *etc*. Further, since the nodes are oblivious to coding, the existing distributed computing infrastructure and software can be directly used without additional control/coordination messaging.

In this paper we focus on a model where nodes become unavailable for the time frame of computation, where a failed node does not recover throughout the duration of

This work was supported in part by NSF grants 1314937 and 1514531.

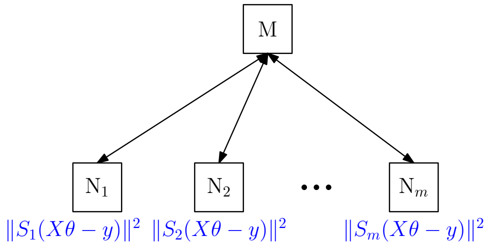

Fig. 1. A distributed optimization network, where m nodes communicate directly with a centralized server. The local nodes compute terms specific to their data (such as gradients), and the central node aggregates such terms and computes simple steps, like small-dimension projections.

the computation. This can also be thought of as a model where slow/straggling nodes are the same ones throughout the computation, and these nodes are ignored by the system. The case with asynchronous/intermittent failures and delays is a natural ongoing extension.

Our main contributions are as follows. First, we derive a general bound on relative objective error for encoding with tight frames, and specialize this to equiangular tight frames[1]. Second, using results from analytic number theory, we obtain a tighter bound for a specific construction with redundancy factor 2, which is constructed using Paley graphs [10]. To the best of our knowledge, this is the first analysis of the this particular tight frame construction in the context of robustness against erasures. We also present an error bound for random coding vectors. Bounds for other constructions with other redundancy factors are possible. Third, we prove a lower bound on the objective error for the special case of unconstrained least squares optimization. Fourth, we numerically demonstrate performance over three problems, two of which use real world datasets, namely, ridge regression, binary support vector machine classification, and low rank approximation. The results show that the Paley construction outperforms uncoded, replication, and random coding approaches.

The rest of the paper is organized as follows: Section II presents our model and metrics of interest, Section III provides our results on encoding with tight frames, Section IV gives lower bounds for general linear encoding, and Section V contains the numerical results on real datasets.

II. Model and Preliminaries

Consider the minimization

$$\min_{\theta \in C} g(\theta) := \min_{\theta \in C} \|X\theta - y\|^2, \tag{1}$$

where $C \subseteq \mathbb{R}^d$ is an arbitrary convex set (that is globally known), $X \in \mathbb{R}^{n \times d}$ is the data matrix, and $y \in \mathbb{R}^n$ is the data vector. We will denote a solution of this optimization as θ^*.

Consider mapping this optimization problem into a distributed computing setup (see Figure 1), where the data variables X_i and y_i are collectively stored across m worker nodes, and a centralized server computes the solution without ever seeing the data itself. Such an architecture is present in most of the popular distributed computing and optimization frameworks [11], [12]. Each worker node has sufficient memory to store $\ell(d+1)$ variables (*i.e.*, ℓ rows of data), where $m\ell \geq n$. We define the redundancy factor $\beta := \frac{m\ell}{n} \geq 1$, which captures the amount of additional storage space available. We consider a linear mapping of the data, where worker node $i \in [m]$ stores $Z_i = S_i \left[X \;\; y\right]$, where $S_i \in \mathbb{R}^{\ell \times n}$ is an encoding matrix. We define $S = \left[S_1^\top \;\; S_2^\top \;\; \ldots S_m^\top\right]^\top$. Note that, by setting $S = I_n$, or $S = \left[I_n \;\; I_n \;\; \ldots\right]^\top$, this framework covers uncoded and repetition schemes as well[2].

We assume that after the data placement, a subset $A \subseteq [m]$ of the nodes are unavailable, and the data stored in the unavailable nodes is assumed to be lost throughout the duration of optimization, where $|A| = e$. We define, for a set $U \subseteq [m]$, $S_U = [S_i]_{i \in U}$, *i.e.*, S_U is the submatrix of S corresponding to the set of nodes U, and $A^c = [m] \backslash A$.

Given a mapping S of the data, the worker nodes directly communicate with the centralized server via (two-way) links with no communication constraints, but cannot communicate with each other. The worker nodes are also oblivious to the encoding (*i.e.*, they do not have access to $\{S_i\}$). These two assumptions imply that the nodes effectively attempt to solve the encoded problem $\min_{\theta \in C} \bar{g}(\theta)$, where

$$\bar{g}(\theta) := \|SX\theta - Sy\|^2 = \sum_{i=1}^{m} \|S_i X\theta - S_i y\|^2 \tag{2}$$

using any distributed optimization algorithm (*e.g.*, batch or stochastic gradient descent, L-BFGS, proximal gradient descent etc.). Since the objective function (2) is a sum of local terms, by having all worker nodes compute, for instance, local gradient terms, and summing them at the centralized server, the centralized solution of (2) can be achieved.

We also assume that the available nodes (A^c) are oblivious to the failed nodes (A), and they operate as if all nodes are available. This assumption, and the fact that the failed nodes (A) are unavailable throughout optimization imply that the effective problem whose solution is reached is

$$\min_{\theta \in C} \widetilde{g}(\theta) := \min_{\theta \in C} \|S_{A^c}(X\theta - y)\|^2. \tag{3}$$

We denote a solution to (3) as $\hat{\theta}(S; X, y; A)$. Given an encoding matrix S, data variables (X, y), and a failure pattern A, the *relative error* $\eta^*(S; X, y; A)$ is defined as the smallest $\eta \geq 1$ such that

$$\|X\hat{\theta} - y\|^2 \leq \eta \|X\theta^* - y\|^2.$$

[1] Performance of frames under erasures have been studied in [7], [8], [9], though not in the context of convex optimization. Further, these works either focus on exact reconstruction, or only one or two erasures, or otherwise do not provide a general error bound for arbitrary tight frames under arbitrary number of erasures.

[2] From a technical standpoint, such linear encoding resembles the sketching technique [6] used to approximate optimization problems by dimensionality reduction. However, sketching uses randomized, short and wide S matrices for dimensionality reduction; we use tall, deterministic S matrices to *increase* the problem dimensions and add redundancy.

For a given S, the *worst-case relative error* is given by

$$\gamma(S,e) := \sup_{X,y} \max_{A:|A|=e} \eta^*(S;X,y;A).$$

Our goal is to design a matrix S such that $\gamma(S,e)$ is minimized and grows slowly with e, *i.e.*, whose worst-case relative error degrades gracefully with increasing number of failed nodes.

III. Encoded Distributed Convex Programs

Intuitively, one would expect a good encoding matrix S to satisfy a number of properties. First, it must contain some form of redundancy in its set of encoding vectors (the rows $s_i^\top$ of S). Second, drawing from the intuition of the channel coding theorem, individual encoding vectors must provide as much independent information as possible. Third, the encoding matrix should not *add* error; that is, when there are no failures, the exact solution must be recoverable, assuming nodes are oblivious to coding. Given such requirements, we turn to *equiangular tight frames* (ETF) as a natural choice of set of encoding vectors. Loosely speaking, ETFs constitute an overcomplete basis for $\mathbb{R}^n$, and whose individual elements are as decorrelated as possible. More formally, a (unit-norm) tight frame for $\mathbb{R}^n$ is a set $\{h_i\}_{i=1}^{n\beta} \subseteq \mathbb{R}^n$ of unit vectors (with $\beta \geq 1$), such that for any $u \in \mathbb{R}^n$,

$$\sum_{i=1}^{n\beta} |\langle h_i, u\rangle|^2 = \beta \|u\|^2. \tag{4}$$

The reader is referred to [13], [10] for more information on frames.

Define the maximal inner product of a tight frame H by

$$\epsilon(H) := \max_{\substack{h_i,h_j\in H\\ i\neq j}} |\langle h_i, h_j\rangle|\,.$$

A tight frame for which $|\langle h_i, h_j\rangle| = \epsilon(H)$ for every $i \neq j$ is called an *equiangular tight frame* (ETF).

Proposition 1 (Welch bound, [14]): Let $H = \{h_i\}_{i=1}^{n\beta}$ be a tight frame. Then $\epsilon(H) \geq \sqrt{\frac{\beta-1}{2n\beta-1}}$. Moreover, equality is satisfied if and only if H is an equiangular tight frame.

Therefore, an ETF minimizes the correlation between its individual elements.

We define the tangent cone of the constraint set at the optimum by

$$\mathcal{K} := \text{clconv}\left\{u \in \mathbb{R}^d : u = t(\theta - \theta^*), t \geq 0, \theta \in C\right\},$$

where clconv denotes closure of the convex hull, and the linearly transformed cone is defined by $X\mathcal{K} := \{Xu : u \in \mathcal{K}\}$. We also define, for a set $\mathcal{U}$, and a symmetric matrix P,

$$\lambda_{\max}^{\mathcal{U}}(P) = \sup_{u\in\mathcal{U},\ \|u\|_2=1} \|Pu\|_2.$$

The case $\lambda_{\max}^{\mathbb{R}^n}(P) = \lambda_{\max}(P)$, the largest eigenvalue of P in absolute value (which is the spectral norm, since P is symmetric).

Our first result bounds the relative error under encoding with tight frames.

Theorem 1: Let S be such that $\{s_i\}_{i=1}^{n\beta}$ is a tight frame over $\mathbb{R}^n$. Then for any encoded optimization problem in the form (3),

$$\eta^*(S;X,y;A) \leq \min_{0\leq c\leq\beta}\left(1 + \frac{2\lambda_{\max}^{X\mathcal{K}}\left(S_A^\top S_A - cI\right)}{\beta - \lambda_{\max}\left(S_A^\top S_A\right)}\right)^2.$$

Corollary 1: Under the setup of Theorem 1,

$$\gamma(S,e) \leq \left(\frac{\beta}{\beta - \max_{A:|A|=e}\left\|S_A^\top S_A\right\|_2}\right)^2.$$

The proofs are given in Appendix A, which relies on techniques from [15], as well as convex optimality conditions and properties of tight frames. Note that the bound only depends on the spectral properties of the *lost* component of the encoding matrix S.

Theorem 1 and Corollary 1 show that when one encodes the data with tight frames, worst-case relative error can be uniformly bounded, and the error depends on the spectral properties of the relevant submatrices S_A of the encoding matrix. We note that (as expected), as the redundancy factor β grows, relative error goes to 1, and when $e = 0$, it is exactly 1, which implies perfect recovery when no failures occur. Note that this is not necessarily true for an arbitrary matrix S whose Gram matrix $S^\top S$ has non-zero eigenvalue spread, including random matrices. We also note that to minimize the error, one must design S such that any possible submatrix S_A has spectral norm close to 1.

Next we prove explicit bounds for *equiangular* tight frames, by bounding the spectral norm of the submatrices S_A. Although these bounds are non-trivial, numerical evidence suggests that tighter bounds may hold.

Theorem 2: If the rows of S form an equiangular tight frame, then for $1 \leq e < \frac{\beta-1}{\alpha(m,n)}$,

$$\gamma(S,e) \leq \left(\frac{\beta}{\beta - 1 - e\alpha(m,n)}\right)^2,$$

where $\alpha(m,n) = \frac{1}{m}\sqrt{\frac{n\beta(\beta-1)}{1-(n\beta)^{-1}}}$.

See Appendix B for proof. For a specific construction obtained by using Paley conference matrices [10], we can in fact prove a tighter result that holds with high probability (under random failures). Let q be a prime number such that $q \equiv 1 \pmod 4$, and let $\mathbb{F}_q$ be the finite field of size q. Consider the graph G_q whose vertices are the elements of $\mathbb{F}_q$, and the elements $a \neq b$ are adjacent if and only if there exists $r \in \mathbb{F}_q$ such that $a - b \equiv r^2 \pmod q$ (in which case $a - b$ is called a *quadratic residue*, and G_q is known as Paley graph). It can be shown that [10] if A_{q+1} is the 0-1 adjacency matrix of the graph formed by combining G_q with an isolated node u, then the matrix $M_{q+1} := \frac{1}{\sqrt{q}}(J_{q+1} - I_{q+1} - 2A_{q+1}) + I_{q+1}$, where J_{q+1} is the all-ones matrix, can be decomposed as

$$M_{q+1} = S_{q+1}S_{q+1}^\top,$$

where the rows of S_{q+1} form an equiangular tight frame with $\epsilon(S_{q+1}) = \frac{1}{\sqrt{q}}$. Using number-theoretic results on multiplicative quadratic residue characters in finite fields (see

Appendix C), we can obtain the following tighter bound for this construction.

Theorem 3: Let $\breve{S}$ be an ETF constructed from Paley graph as above, where $q+1 = 2n$ (so that redundancy factor $\beta = 2$). Let $S = P\breve{S}$, where P is a random permutation matrix that is drawn uniformly random over all $(2n)!$ permutation matrices. Let A be uniformly random over all cardinality-e subsets of $[m]$. Then for $1 \leq e < \left(\frac{1}{c\tilde{\alpha}(m,n)}\right)^{4/3}$ and for any $c > 1$,

$$\mathbb{P}\left(\eta(S; X, y; A) > \left(\frac{2}{1 - ce^{3/4}\tilde{\alpha}(m,n)}\right)^2\right) \leq \frac{1}{c^4},$$

where $\tilde{\alpha}(m,n) := \sqrt{\frac{2}{m-\frac{1}{\ell}}}\left(\frac{2n}{m}\right)^{1/4}$.

To the best of our knowledge, Theorem 3 is the first analysis of the erasure-robustness of Paley ETFs. This result shows that if we scale the number of nodes m faster than $n^{\frac{1}{3}}$, then the error is small with high probability, even under a large number of node failures. In fact, based on numerical evidence, we believe the following, even tighter, deterministic bound holds for this construction.

Conjecture 1: If S is an ETF constructed from Paley graph as above, where $q+1 = 2n$, then for $1 \leq e < \frac{1}{\tilde{\alpha}^2(m)}$,

$$\gamma(S, e) \leq \left(\frac{2}{1 - \sqrt{e}\tilde{\alpha}(m)}\right)^2,$$

where $\tilde{\alpha}(m) := \frac{c}{\sqrt{m}}$ for a universal constant c.

Note that there is no dependence on n in this bound.

Random coding: Another natural approach in designing S could be choosing its elements i.i.d. random, *e.g.*, with Gaussian entries. In particular, using results from [15], and the scaling behavior of singular values of i.i.d. Gaussian matrices [16], it can be shown that the following holds (the details are in Appendix D).

Proposition 2: For fixed $\beta = \frac{m\ell}{n}$, consider a family of encoding matrices $S_m \in \mathbb{R}^{m\ell \times \frac{m\ell}{\beta}}$, indexed by the number of worker nodes m. Choose all entries of S_m i.i.d. from $N\left(0, \frac{1}{n}\right)$. Denote the relative error for m machines as $\eta_m^*(S_m; X, y; A)$, for any A with $|A| = e < m\frac{\beta-1}{\beta}$. Then, for any (X, y),

$$\lim_{m\to\infty} \eta_m^*(S_m; X, y; A) \leq \left(\frac{\sqrt{\beta\left(1 - \frac{e}{m}\right)} + 1}{\sqrt{\beta\left(1 - \frac{e}{m}\right)} - 1}\right)^4.$$

Note that random coding can achieve a bound independent of n as well, albeit asymptotically. In practice, however, we observe that the spectral norm of submatrices of Paley ETF grows slower than those of i.i.d. random matrices, and thus Paley ETF achieves a slightly tighter bound on relative error for finite data, as claimed in Conjecture 1, and further evidenced in the results of the next section.

IV. Lower Bound for Unconstrained Optimization

Given the results of Section III, one may wonder how they compare with the performance of other possible encoding techniques. In this section, we derive a lower bound on the relative error for unconstrained optimization ($C = \mathbb{R}^d$) for arbitrary linear encoding. The bound is not necessarily tight, but it still provides insight into how one should design the encoding matrix.

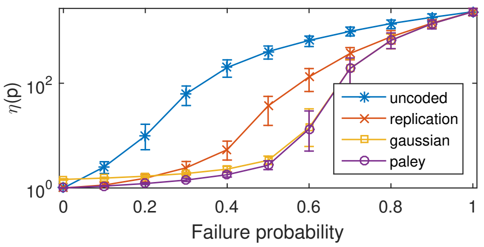

Fig. 2. Performance for ridge regression, where X is 1000×750 and $\mu = 0.1$. There are 750 processors and S has 2000 rows.

Theorem 4: For any encoding matrix S, worst-case relative error for unconstrained optimization is lower bounded by

$$\gamma(S, e) \geq \frac{1}{4}(1 + \max_{A:|A|=e} \kappa(S_{A^c}))^2$$

where $\kappa(Q)$ is the condition number of matrix Q.

The proof is provided in Appendix E, which is based on constructing an adversarial data pair (X, y) for any given encoding matrix S. Theorem 4 implies that in order to control the error, one needs to design the encoding matrix so that any relevant submatrix S_{A^c} is well-conditioned, which is similar to the restricted isometry condition in compressed sensing [17].

V. Numerical Results

We explore three machine learning problems, two of which use real world datasets. In each example, we compare four cases: uncoded ($S = I_n$), replication code, Gaussian ($S_{ij} \sim \mathcal{N}(0,1)$), and Paley ETF. The redundancy factor $\beta = 2$ in each case except the uncoded one. In the simulations, we consider probabilistic availability of the nodes, where each node independently fails with probability p. In each case we plot relative error ($\eta(p)$, representing relative error at failure probability p) over 100 trials with different failure patterns, with error bars at a 95% interval.

A. Ridge regression

The encoded ridge regression problem solves

$$\underset{\theta}{\text{minimize}} \quad \|S(X\theta - y)\|_2^2 + \mu\|\theta\|_2^2, \tag{5}$$

where $\mu > 0$ is a regularization parameter. The rows of X and y represent data feature vectors and labels respectively, and the entries of the solution θ^* are the feature regressors.

Figure 2 shows the relative error performance with respect to failure probability, where $y = Xz + n$ and each element of X, y, and n is drawn independently from a Gaussian distribution. The data matrix X is 1000×750 and the generated encoding matrices have 1000 (uncoded), 2000 (replication,

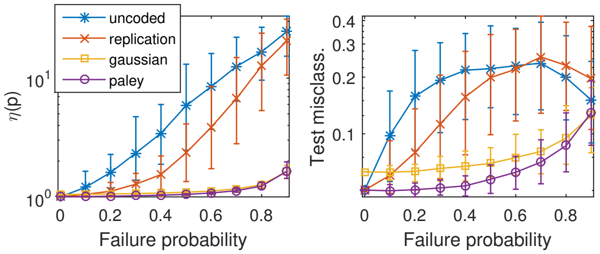

Fig. 3. Performance for solving SVM on reduced MNIST set for 4 vs. 9 disambiguation. Here X is 1000×784 and $\mu = 0.1$. There are 500 processors.

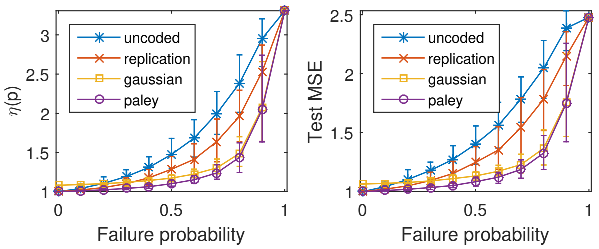

Fig. 4. Performance for solving the matrix completion problem with the subsampled movielens dataset. Here, X is 2757×7448, and $\tau = 100$. There are 100 processors and S has twice the number of rows as X.

Gaussian, Paley[3]) rows. The problem is solved using gradient descent, where each worker node computes gradient terms corresponding to their own data and the central node only performs the aggregation and descent step.

B. Binary SVM classification

The MNIST dataset contains 28×28 binary images for handwritten digits 0-9 [18]. We attempt to disambiguate 4's from 9's using binary support vector machines, by solving the reformulation suggested by [15, §3.4]:

$$\begin{array}{ll} \underset{\theta}{\text{minimize}} & \|W^T \mathbf{diag}(d)\theta\|_2^2 + \mu\|\theta\|_2^2 = \|X\theta\|_2^2 \\ \text{subject to} & \sum_i \theta_i = 1, \quad \theta_i \geq 0, \forall i. \end{array} \tag{6}$$

The rows W_i are ith vectorized binary images (demeaned), and $d_i \in \{1, -1\}$ indicates if the ith sample is a 4 or 9. The objective can be reformulated with $X = [\mathbf{diag}(d)W, I]^T$, and the encoded problem has objective $\|SX\theta\|_2^2$.

We reduce the MNIST train and test dataset to only the digits 4 and 9, and additionally only use the first 1000 train samples ($W \in \mathbb{R}^{1000 \times 784}$). Fig. 3 shows the relative error performance, where (6) is solved using FISTA [19], where the worker nodes evaluate gradients and the centralized server aggregates terms and computes the projection on the simplex.

C. Low rank approximation

The movielens ml-100k dataset [20] contains recommendations of users for movies. The task is, given ratings in a training set, predict the ratings in a separate test set. Given rating matrix R, where R_{ij} is the rating user i provided movie j (if exists in the training set), and find the nearest low rank approximate matrix completion of R. The following is an encoded version of a popular convex approximation of the rank-constrained matrix completion problem:

$$\begin{array}{ll} \underset{\Theta}{\text{minimize}} & \|SX\mathbf{vec}(\Theta - R)\|_F^2 \\ \text{subject to} & \|\Theta\|_* \leq \tau. \end{array} \tag{7}$$

Here, $\|Z\|_*$ is the nuclear norm (sum of the singular values of Z) and serves as a convex proxy for rank. The matrix X is such that $X\mathbf{vec}(R)$ selects only the provided ratings R_{ij}.

We subsample the movielens dataset to leave only users and movies that contribute the most ratings, resulting in 133 users and 56 movies, with 5,514 provided ratings evenly split between train and test sets. (Resulting X is 2757×7448.) (7) is solved using FISTA [19], with $\tau = 100$. Figure 4 shows relative error results and the mean squared error in test ratings, defined as $\frac{1}{|T|}\sum_{(i,j)\in T}((R_{\text{test}})_{ij} - X_{ij})^2$ where R_{test} is the test ratings matrix and T contains the (user, movie) pairs included in the test set.

In all three examples, it is clear that coding increases robustness in the presence of large numbers of node failures, both in the relative error of the objective and in test error metrics on real datasets. The tightness of the Paley frames is also observed; in all cases there is no degradation when no nodes fail, which is not true when using random encoding matrices.

References

[1] J. Dean and L. A. Barroso, "The tail at scale," *Communications of the ACM*, vol. 56, no. 2, pp. 74–80, 2013.

[2] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Fundamental tradeoff between computation and communication in distributed computing," in *Information Theory (ISIT), 2016 IEEE International Symposium on*, pp. 1814–1818, IEEE, 2016.

[3] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," in *Information Theory (ISIT), 2016 IEEE International Symposium on*, pp. 1143–1147, IEEE, 2016.

[4] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding," *ML Systems Workshop (MLSyS), NIPS*, 2016.

[5] S. Dutta, V. Cadambe, and P. Grover, "Short-dot: Computing large linear transforms distributedly using coded short dot products," in *Advances In Neural Information Processing Systems*, pp. 2092–2100, 2016.

[6] M. W. Mahoney, "Randomized algorithms for matrices and data," *Foundations and Trends® in Machine Learning*, vol. 3, no. 2, pp. 123–224, 2011.

[7] V. K. Goyal, J. Kovačević, and J. A. Kelner, "Quantized frame expansions with erasures," *Applied and Computational Harmonic Analysis*, vol. 10, no. 3, pp. 203–233, 2001.

[8] R. B. Holmes and V. I. Paulsen, "Optimal frames for erasures," *Linear Algebra and its Applications*, vol. 377, pp. 31–51, 2004.

[9] P. G. Casazza and J. Kovačević, "Equal-norm tight frames with erasures," *Advances in Computational Mathematics*, vol. 18, no. 2-4, pp. 387–430, 2003.

[10] T. Strohmer and R. W. Heath, "Grassmannian frames with applications to coding and communication," *Applied and computational harmonic analysis*, vol. 14, no. 3, pp. 257–275, 2003.

[11] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[3] Since Paley ETF has size $(q+1) \times (q+1)/2$ for prime q, we take the smallest prime s.t. $q \equiv 1 \pmod 4$ (in this case, 2017) larger than the required dimension, and take an arbitrary submatrix that matches the required dimensions. The error due to this subsampling is negligible.

[12] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pp. 2–2, USENIX Association, 2012.

[13] I. Daubechies, *Ten lectures on wavelets*. SIAM, 1992.

[14] L. Welch, "Lower bounds on the maximum cross correlation of signals (corresp.)," *IEEE Transactions on Information theory*, vol. 20, no. 3, pp. 397–399, 1974.

[15] M. Pilanci and M. J. Wainwright, "Randomized sketches of convex programs with sharp guarantees," *IEEE Transactions on Information Theory*, vol. 61, no. 9, pp. 5096–5115, 2015.

[16] J. W. Silverstein, "The smallest eigenvalue of a large dimensional wishart matrix," *The Annals of Probability*, pp. 1364–1368, 1985.

[17] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE transactions on information theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[18] Y. LeCun, C. Cortes, and C. J. Burges, "The MNIST database of handwritten digits," 1998.

[19] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[20] J. Riedl and J. Konstan, "Movielens dataset," 1998.

[21] T. M. Apostol, *Introduction to analytic number theory*. Springer Science & Business Media, 2013.

[22] S. Geman, "A limit theorem for the norm of random matrices," *The Annals of Probability*, pp. 252–261, 1980.

Appendix A
Proofs of Theorem 1 and Corollary 1

The proof is based on a variation of the proof of the main result in [15]; however, unlike the proof therein, we make use of the properties of tight frames.

Fix a failure pattern A. We first note that since the rows of S form a tight frame, $S^\top S = \beta I_n$. Recalling that $s_i^\top$ is the ith row of S,

$$\begin{aligned} S_A^\top S_A &= \sum_{i\in A} s_i s_i^\top = \sum_{i=1}^{n\beta} s_i s_i^\top - \sum_{i\notin A} s_i s_i^\top \\ &= S^\top S - S_{A^c}^\top S_{A^c} = \beta I_n - S_{A^c}^\top S_{A^c}. \end{aligned} \tag{8}$$

Denoting the minimum and maximum eigenvalues of a matrix by $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ respectively, and using (8), any unit vector u satisfies

$$\|S_{A^c}u\|^2 \geq \lambda_{\min}\left(S_{A^c}^\top S_{A^c}\right) = \beta - \lambda_{\max}\left(S_A^\top S_A\right). \tag{9}$$

Defining $e = \hat{\theta} - \theta^*$, we have

$$\|X\hat{\theta} - y\| \leq \left(1 + \frac{\|Xe\|}{\|X\theta^* - y\|}\right)\|X\theta^* - y\|,$$

by triangle inequality. Therefore

$$\eta(S; X, y; \alpha) \leq \left(1 + \frac{\|Xe\|}{\|X\theta^* - y\|}\right)^2. \tag{10}$$

For any $0 \leq c \leq \beta$, consider

$$\begin{aligned} \|Xe\|^2 &\overset{(a)}{\leq} \frac{\|S_{A^c}Xe\|^2}{\beta - \lambda_{\max}\left(S_A^\top S_A\right)} \\ &\overset{(b)}{\leq} -2\frac{e^\top X^\top S_{A^c}^\top S_{A^c}(X\theta^* - y)}{\beta - \lambda_{\max}\left(S_A^\top S_A\right)} \\ &= -2\frac{e^\top X^\top \left(S_{A^c}^\top S_{A^c} - (\beta - c)\, I\right)(X\theta^* - y)}{\beta - \lambda_{\max}\left(S_A^\top S_A\right)} \\ &\quad - \frac{2\,(\beta - c)}{\beta - \lambda_{\max}} e^\top X^\top (X\theta^* - y) \\ &\overset{(c)}{\leq} -2\frac{e^\top X^\top \left(S_{A^c}^\top S_{A^c} - (\beta - c)\, I\right)(X\theta^* - y)}{\beta - \lambda_{\max}\left(S_A^\top S_A\right)} \\ &\overset{(d)}{=} 2\frac{e^\top X^\top \left(S_A^\top S_A - cI\right)(X\theta^* - y)}{\beta - \lambda_{\max}\left(S_A^\top S_A\right)} \\ &\overset{(e)}{\leq} 2\frac{\left\|e^\top X^\top \left(S_A^\top S_A - cI\right)\right\|}{\beta - \lambda_{\max}\left(S_A^\top S_A\right)}\|X\theta^* - y\| \\ &\overset{(f)}{\leq} \frac{2\lambda_{\max}^{X\mathcal{K}}\left(S_A^\top S_A - cI\right)}{\beta - \lambda_{\max}\left(S_A^\top S_A\right)}\|Xe\|\|X\theta^* - y\|, \end{aligned}$$

where (a) follows by (9); (b) follows by re-arranging $\|S_{A^c}(X\hat{\theta} - y)\|^2 \leq \|S_{A^c}(X\theta^* - y)\|^2$, which is true because of the optimality of $\hat{\theta}$ for the encoded problem; (c) follows by the convex optimality condition

$$\langle X^\top (X\theta^* - y), e\rangle = \langle \nabla g(\theta^*), \hat{\theta} - \theta^*\rangle \geq 0;$$

(d) follows by (8); (e) follows by Cauchy-Schwarz inequality; and (f) follows by the definition of $\lambda_{\max}^{X\mathcal{K}}$, and the fact that $\hat{\theta}$ is feasible, so $e \in \mathcal{K}$. This bound, together with (10), implies Theorem 1 by minimizing over all possible choices of c.

To prove Corollary 1, first note that the bound is maximized when $X\mathcal{K}$ contains the eigenvector of $\left(S_A^\top S_A - cI\right)$ corresponding to the largest eigenvalue. Choose X to map an arbitrary $e \in \mathcal{K}$ to this eigenvector, which implies $\lambda_{\max}^{X\mathcal{K}}(S_A^\top S_A - cI) = \lambda_{\max}(S_A^\top S_A - cI)$ (recall that $\lambda_{\max}$ refers to the maximum *absolute value* of the eigenvalues, hence equivalent to operator norm for any symmetric matrix). Further choose $c = \frac{1}{2}\lambda_{\max}(S_A^\top S_A)$ to get

$$\begin{aligned} \gamma(S, e) &\leq \min_{0\leq c\leq\beta}\max_{|A|=e}\left(1 + \frac{2\lambda_{\max}\left(S_A^\top S_A - cI\right)}{\beta - \lambda_{\max}\left(S_A^\top S_A\right)}\right)^2 \\ &= \max_{|A|=e}\left(1 + \frac{2\lambda_{\max}\left(S_A^\top S_A - \frac{1}{2}\lambda_{\max}(S_A^\top S_A)I\right)}{\beta - \lambda_{\max}\left(S_A^\top S_A\right)}\right)^2 \\ &\overset{(g)}{=} \max_{|A|=e}\left(1 + \frac{\lambda_{\max}\left(S_A^\top S_A\right)}{\beta - \lambda_{\max}\left(S_A^\top S_A\right)}\right)^2 \\ &= \max_{|A|=e}\left(\frac{\beta}{\beta - \lambda_{\max}\left(S_A^\top S_A\right)}\right)^2, \end{aligned}$$

where (g) follows by the fact that all eigenvalues of $S_A^\top S_A$ are between 0 and $\lambda_{\max}(S_A^\top S_A)$ and thus the absolute values of all eigenvalues of $S_A^\top S_A - \frac{1}{2}\lambda_{\max}(S_A^\top S_A)I$ are upper bounded by $\frac{1}{2}\lambda_{\max}(S_A^\top S_A)$.

Appendix B
Proof of Theorem 2

First we would like to bound $\left\|S_A S_A^\top - I_{e\ell}\right\|_2$. Note that the (i, j)th element of $S_A S_A^\top$ is given by $\langle s_i, s_j\rangle$ for $i \neq j$, where $s_i^\top$ is the ith row of S_A, and the diagonal of $S_A S_A^\top - I_{e\ell}$ consists of zeros. Since S is equiangular, Proposition 1 implies that $|\langle s_i, s_j\rangle| = \sqrt{\frac{\beta-1}{n\beta-1}}$. Then by Gershgorin circle theorem, all eigenvalues $\{\lambda_k\}$ of $S_A S_A^\top - I_{e\ell}$ satisfy

$$|\lambda_k| \leq \sum_{j=1}^{e\ell} |\langle s_i, s_j \rangle| = e\ell \sqrt{\frac{\beta - 1}{n\beta - 1}},$$

which, using the fact $\ell = \frac{n\beta}{m}$ implies,

$$\left\| S_A S_A^\top - I_{e\ell} \right\|_2 \leq \frac{e}{m} \sqrt{\frac{n\beta(\beta-1)}{1 - \frac{1}{n\beta}}}.$$

Using triangle inequality,

$$\left\| S_A^\top S_A \right\|_2 = \left\| S_A S_A^\top \right\|_2 \leq 1 + \frac{e}{m} \sqrt{\frac{n\beta(\beta-1)}{1 - \frac{1}{n\beta}}}.$$

Plugging in this bound in Corollary 1 gives the desired result.

Appendix C
Proof of Theorem 3

Recall that by construction, $2n - 1 = q$ is a prime such that $q \equiv 1 \pmod 4$. For any row index $1 \leq i \leq q$, define $\kappa(i)$ as the index of the node row i of S corresponds to, *i.e.*, $\kappa(i) := \lceil \frac{i}{\ell} \rceil$. Further define $\pi : [2n] \to [2n]$ to be a random permutation of the integers $\{1, \ldots, 2n\}$, which is uniform over each of the $(2n)!$ realizations.

Let J_i be the 0-1 indicator variable denoting whether node i is unavailable, (*i.e.*, $J_i = 1$ if and only if $i \in A$), and $J := \{J_i\}_{i=1}^m$. Given e, we assume J takes uniformly at random one of the $\binom{m}{e}$ vector values consisting of e 1's, and $m - e$ 0's. Note that J_i and J_j are not independent for $i \neq j$.

Given a finite field $\mathbb{F}_q$, $a \in \mathbb{F}_q$ is called a *quadratic residue* if there exists $r \in \mathbb{F}_q$ such that $a \equiv r^2 \pmod q$. Construct the matrix $L \in \{-1, 0, 1\}^{2n \times 2n}$ such that

$$L_{ij} = \begin{cases} \chi(i - j), & 1 \leq i, j \leq q \\ \mathbb{I}_{i \neq j}, & \text{if } i = q + 1 \text{ or } j = q + 1 \end{cases}$$

where χ is the quadratic residue character in $\mathbb{F}_q$, defined by

$$\chi(x) = \begin{cases} 0, & \text{if } x = 0, \\ 1, & \text{if } x \neq 0 \text{ is a quadratic residue in } \mathbb{F}_q, \\ -1, & \text{otherwise.} \end{cases}$$

In the above definition, we have assumed that the $(q+1)$th index corresponds to the isolated node appended to the Paley graph.

Characters are important objects of study in analytic number theory (see, *e.g.*, [21] for more information on characters). In particular, quadratic residue character χ is a *multiplicative* character, satisfying the following properties, which can be easily verified:

1) $\chi(1) = 1$,
2) For $a, b \in \mathbb{F}_q$, $\chi(a)\chi(b) = \chi(ab)$,
3) For $a \in \mathbb{F}_q$, $\chi(a) = \chi(a^{-1})$.

Proposition 3 ([21]): Let q be an odd prime. The quadratic residue character χ over $\mathbb{F}_q$ satisfies

$$\sum_{a \in \mathbb{F}_q} \chi(a) = 0.$$

Define

$$\bar{L} := \left[L_{ij} J_{\kappa(\pi(i))} J_{\kappa(\pi(j))} \right]_{i,j}.$$

Note that the matrix

$$\sqrt{q} \left(S_A S_A^\top - I \right) \tag{11}$$

is identical to the realization of $\bar{L}$ corresponding to J such that $J_i = 1 \Leftrightarrow i \in A$, up to padding with zeroes. Therefore, they have the same spectrum and the problem reduces to characterizing the expected spectral norm of $\bar{L}$.

We will prove the following lemma.

Lemma 1: Let $a, b \in \mathbb{F}_q$. Then

$$\sum_{x \in \mathbb{F}_q} \chi(a - x)\chi(b - x) = \left(-1 + q\mathbb{I}_{a=b} \right).$$

Proof: The case $a = b$ easily follows by the fact that

$$\sum_{x \in \mathbb{F}_q} \chi(a - x)\chi(a - x) = \sum_{x \in \mathbb{F}_q} \mathbb{I}_{a \neq x} = q - 1.$$

If $a \neq b$, using properties of $\chi(\cdot)$,

$$\begin{aligned} \sum_{x \in \mathbb{F}_q} \chi(a - x)\chi(b - x) &= \sum_{x \neq a,b} \chi(a - x)\chi(b - x) \\ &= \sum_{x \neq a,b} \chi(a - x)\chi\left((b - x)^{-1}\right) = \sum_{x \neq a,b} \chi\left(\frac{a - x}{b - x}\right) \\ &= \sum_{x \neq a,b} \chi\left(1 + \frac{a - b}{b - x}\right) \overset{\text{(a)}}{=} \sum_{y \neq 0,1} \chi(y) \overset{\text{(b)}}{=} -\chi(1) = -1, \end{aligned}$$

which completes the proof. (a) follows because in $\mathbb{F}_q$ every non-zero element has a unique multiplicative inverse, hence the argument of the character will take every value except 0 (since $x \neq a$) and 1 (since $a \neq b$); (b) follows by Proposition 3. ■

Now, consider

$$\begin{aligned} &\mathbb{E}\left[\operatorname{tr}\left(\bar{L}^4\right) \right] \\ &= \mathbb{E}\left[\sum_{i_1, \ldots, i_4} L_{i_1 i_2} L_{i_2 i_3} L_{i_3 i_4} L_{i_4 i_1} J_{\kappa(\pi(i_1))} \ldots J_{\kappa(\pi(i_4))} \right] \\ &= \sum_{i_1, \ldots, i_4} L_{i_1 i_2} L_{i_2 i_3} L_{i_3 i_4} L_{i_4 i_1} \mathbb{E}\left[J_{\kappa(\pi(i_1))} \ldots J_{\kappa(\pi(i_4))} \right]. \end{aligned}$$

Note that, since π is uniformly random, we have

$$\mathbb{E}\left[J_{\kappa(\pi(i_1))} \ldots J_{\kappa(\pi(i_4))} \right] = \frac{\binom{e\ell}{s}}{\binom{m\ell}{s}} \leq \left(\frac{e}{m} \right)^s,$$

where s is the number of unique elements in the tuple (i_1, i_2, i_3, i_4). Therefore

$$\begin{aligned} \mathbb{E}\left[\operatorname{tr}\left(\bar{L}^4\right) \right] &\leq \sum_{s=1}^{4} \left(\frac{e}{m} \right)^s \sum_{\substack{i_1, \ldots, i_4: \\ \{i_1, i_2, i_3, i_4\} = s}} L_{i_1 i_2} L_{i_2 i_3} L_{i_3 i_4} L_{i_4 i_1} \\ &=: \sum_{s=1}^{4} \left(\frac{e}{m} \right)^s \phi(s), \end{aligned}$$

where we have defined the inner sum as $\phi(s)$. First, note that $\phi(1) = 0$ by the fact that this would require all i_j to be equal, and $L_{i_j i_j} = 0$ by definition. Next, consider

$$\begin{aligned}\phi(2) &= \sum_{\substack{i_1,\ldots,i_4: \\ \{i_1,i_2,i_3,i_4\}=2}} L_{i_1 i_2} L_{i_2 i_3} L_{i_3 i_4} L_{i_4 i_1} \\ &= \sum_{a\neq b} L_{ab} L_{ba} L_{ab} L_{ba} = \sum_{a \neq b} L_{ab}^4 = q(q+1)\end{aligned}$$

by the fact that L is symmetric and all the off-diagonal elements are ±1. Then

$$\begin{aligned}\phi(3) &= \sum_{\substack{i_1,\ldots,i_4: \\ \{i_1,i_2,i_3,i_4\}=3}} L_{i_1 i_2} L_{i_2 i_3} L_{i_3 i_4} L_{i_4 i_1} \\ &= \sum_{\substack{a\neq b \\ a\neq c \\ b \neq c}} L_{ab} L_{bc} L_{cb} L_{ba} + \sum_{\substack{a\neq b \\ a\neq c \\ b \neq c}} L_{ab} L_{ba} L_{ac} L_{ca} \\ &= \sum_{\substack{a\neq b \\ a\neq c \\ b \neq c}} L_{ab}^2 L_{bc}^2 + \sum_{\substack{a\neq b \\ a\neq c \\ b \neq c}} L_{ab}^2 L_{ac}^2 = 2(q+1)q(q-1),\end{aligned}$$

similarly by the symmetry and unit modulus of the elements of L. Now,

$$\begin{aligned}\sum_{s=1}^{4} \phi(s) &= \sum_{i_1,\ldots,i_4} L_{i_1 i_2} L_{i_2 i_3} L_{i_3 i_4} L_{i_4 i_1} \\ &= \sum_{i_1, i_3} \left(\sum_{i_2} L_{i_1 i_2} L_{i_2 i_3} \right) \left(\sum_{i_4} L_{i_3 i_4} L_{i_4 i_1} \right) \\ &= q^2 \sum_{i_1, i_3} \mathbb{I}_{i_1 = i_3} = q^2 (q+1),\end{aligned}$$

where the third equality follows by Lemma 1 and the definition of L_{ij}, which implies

$$\sum_{j=1}^{q+1} L_{ij} L_{jk} = q \mathbb{I}_{i=k}.$$

The above results then imply

$$\begin{aligned}\phi(4) &= q^2(q+1) - 2(q+1)q(q-1) - q(q+1) \\ &= -q(q+1)(q-1).\end{aligned}$$

Hence,

$$\begin{aligned}\mathbb{E}\left[\text{tr}\left(\bar{L}^4\right)\right] &= \sum_{s=1}^{4} \left(\frac{e}{m}\right)^s \phi(s) \\ &= \left(\frac{e}{m}\right)^2 q(q+1) \left(1 + \frac{e}{m}(q-1)\left(2 - \frac{e}{m}\right)\right) \\ &\leq 4 \left(\frac{e}{m}\right)^3 (q+1)^3.\end{aligned}$$

Then, defining λ_i to be the ith largest eigenvalue of $\bar{L}$, for any $a > 0$,

$$\begin{aligned}\mathbb{P}\left(\max_i |\lambda_i| > a\right) &= \mathbb{P}\left(\max_i |\lambda_i|^4 > a^4\right) \\ &\leq \mathbb{P}\left(\sum_i |\lambda_i|^4 > a^4\right) = \mathbb{P}\left(\text{tr}\left(\bar{L}^4\right) > a^4\right) \\ &\overset{(a)}{\leq} \frac{\mathbb{E}\left[\text{tr}\left(\bar{L}^4\right)\right]}{a^4} \leq \frac{4\left(\frac{e}{m}\right)^3 (q+1)^3}{a^4},\end{aligned}$$

where (a) is by Markov inequality. Therefore, setting

$$a = c\sqrt{2}\left(\frac{e}{m}\right)^{3/4} (q+1)^{3/4} = c\sqrt{2}\,(e\ell)^{3/4}$$

for some constant $c > 0$, using that for A uniformly random among all $\binom{m}{e}$ possible e failures,

$$\lambda_{\max}\left(S_A^\top S_A\right) = \lambda_{\max}\left(S_A S_A^\top\right) = 1 + \frac{1}{\sqrt{q}} \lambda_{\max}\left(\bar{L}\right),$$

we get

$$\mathbb{P}\left(\lambda_{\max}\left(S_A^\top S_A\right) > 1 + c\sqrt{\frac{2}{m - \frac{1}{\ell}}} e^{3/4} \ell^{1/4}\right) \leq \frac{1}{c^4},$$

which directly implies the result using Corollary 1.

Appendix D
Proof of Proposition 2

We will use the following result (slightly loosened and rephrased in our notation).

Lemma 2 ([15], Lemma 1): For any $c > 0$,

$$\eta\left(S_m; X, y; A\right) \leq \left(1 + 2\frac{\|S_A^\top S_A - cI\|_2}{\lambda_{\min}\left(S_A^\top S_A\right)}\right)^2.$$

Using the results from [16], [22], we know that

$$\begin{aligned}\lambda_{\max}\left(\left(S_m\right)_A^\top \left(S_m\right)_A\right) &\to \left(\sqrt{\beta\left(1 - \frac{e}{m}\right)} + 1\right)^2 \\ \lambda_{\min}\left(\left(S_m\right)_A^\top \left(S_m\right)_A\right) &\to \left(\sqrt{\beta\left(1 - \frac{e}{m}\right)} - 1\right)^2,\end{aligned}$$

almost surely as $m \to \infty$. Plugging these in Lemma 2, and using $c = 1 + \beta\left(1 - \frac{e}{m}\right)$, we get the desired result.

Appendix E
Proof of Theorem 4

Given an S, we will construct a data pair (X, y) so that the quantity

$$\frac{\|X\hat{\theta} - y\|^2}{\|X\theta^* - y\|^2}$$

is maximized, where we choose (X, y) so that $\|X\theta^* - y\|^2 > 0$ by design, so the above is well-defined.

To this end, let us first fix θ^*, and assume $y = X\theta^* + r$, where $r^\top X = 0$, by the optimality condition. We can equivalently construct the pair (X, r). Then the relative error can be written as

$$\begin{aligned}\frac{\|X\hat{\theta} - y\|^2}{\|X\theta^* - y\|^2} &= \frac{\|X\left(\hat{\theta} - \theta^*\right) + r\|^2}{\|r\|^2} \overset{(a)}{=} 1 + \frac{\|X\left(\hat{\theta} - \theta^*\right)\|^2}{\|r\|^2} \\ &\overset{(b)}{=} 1 + \frac{\|X\left(X^\top S_{A^c}^\top S_{A^c} X\right)^{-1} X^\top S_{A^c}^\top S_{A^c} y - X\theta^*\|^2}{\|r\|^2} =\end{aligned}$$

$$1 + \frac{\|X\left(X^\top S_{A^c}^\top S_{A^c} X\right)^{-1} X^\top S_{A^c}^\top S_{A^c}(X\theta^* + r) - X\theta^*\|^2}{\|r\|^2}$$
$$= 1 + \frac{\|X\left(X^\top S_{A^c}^\top S_{A^c} X\right)^{-1} X^\top S_{A^c}^\top S_{A^c} r\|^2}{\|r\|^2}$$

where (a) follows by the fact that $r^\top X = 0$, and (b) follows by plugging in the analytic expression for $\hat{\theta} = \left(S_{A^c}X\right)^\dagger \left(S_{A^c}y\right)$. Let $S_{A^c}^\top S_{A^c} = Q^\top \Lambda Q$ be the eigendecomposition of $S_{A^c}^\top S_{A^c}$, and define $Z = QX$ and $t = Qr$, where we reduced the problem to constructing (Z, t). Then

$$\frac{\|X\hat{\theta} - y\|^2}{\|X\theta^* - y\|^2} \overset{\text{(a)}}{=} 1 + \frac{\|QX\left(X^\top S_{A^c}^\top S_{A^c} X\right)^{-1} X^\top S_{A^c}^\top S_{A^c} r\|^2}{\|Qr\|^2}$$
$$= 1 + \frac{\|Z\left(Z^\top \Lambda Z\right)^{-1} Z^\top \Lambda t\|^2}{\|t\|^2}$$

where (a) follows by the fact that ℓ_2 norm is invariant under orthogonal transformations. Note that since we require $r^\top X = 0$, we have $t^\top Z = 0$. Therefore we set $t = (I - ZZ^\dagger)v$, where there is no constraint on v. Plugging in this value for t and simplifying, and also using the non-expansiveness of the projection, which implies $\|v\|^2 \geq \|t\|^2$, we have

$$\sup_{X,y} \frac{\|X\hat{\theta} - y\|^2}{\|X\theta^* - y\|^2}$$
$$\geq \sup_{Z,v} \left(1 + \frac{\|Z\left(Z^\top \Lambda Z\right)^{-1} Z^\top \Lambda v\|^2 - \|U^\top v\|^2}{\|v\|^2}\right)$$
$$= \sup_{Z,v} \frac{\|Z\left(Z^\top \Lambda Z\right)^{-1} Z^\top \Lambda v\|^2}{\|v\|^2},$$

where U is a $n \times d$ matrix with orthonormal columns, whose columns span the column space of Z. In the last equality, we have used the fact that U is orthogonal.

Now, note that we can assume, without loss of generality, $S_{A^c}^\top S_{A^c}$ is positive definite, since otherwise we can construct (X, y) with unbounded error, by choosing columns of X in the eigenspace of $S_{A^c}^\top S_{A^c}$ associated with zero eigenvalues. Therefore, we can assume Λ is invertible. Define $B = \Lambda^{1/2}Z$, and $P = B\left(B^\top B\right)^{-1} B^\top$ to be the projection matrix on the range space of B. We pick an X such that

$$P = \begin{bmatrix} \frac{1}{2} & 0^\top & \frac{1}{2} \\ 0 & \widetilde{P} & 0 \\ \frac{1}{2} & 0^\top & \frac{1}{2} \end{bmatrix}$$

where 0 is the 0-vector and $\widetilde{P}$ is some other idempotent matrix of the appropriate size. Then P is an appropriate projection matrix for the choice of B as

$$B = \begin{bmatrix} 0^\top & \frac{1}{2} \\ \widetilde{B} & 0 \\ 0^\top & \frac{1}{2} \end{bmatrix}, \quad \tilde{P} = \widetilde{B}\left(\widetilde{B}^\top \widetilde{B}^\top\right)^{-1} \widetilde{B}^\top.$$

We additionally pick $v = \alpha[1, 0, \ldots, 0]^\top$ for any scalar α. Then, denoting with λ_i the ith largest eigenvalue in Λ,

$$\sup_{X,y} \frac{\|X\hat{\theta} - y\|^2}{\|X\theta^* - y\|^2} \geq \sup_{B,v} \frac{\|\Lambda^{-1/2}B\left(B^\top B\right)^{-1} B^\top \Lambda^{1/2} v\|^2}{\|v\|^2}$$

$$= \left(\frac{\lambda_1^{1/2} + \lambda_n^{1/2}}{2\lambda_n^{1/2}}\right)^2 = \frac{1}{4}\left(1 + \kappa(S_{A^c})\right)^2$$

where $\kappa(S_{A^c}) = \frac{\sqrt{\lambda_1}}{\sqrt{\lambda_n}}$ is the condition number of S_{A^c}.