Information and Inference: A Journal of the IMA (2021) **10**, 231–260
doi:10.1093/imaiai/iaaa026
Advance Access publication on 17 October 2020

Fundamental resource trade-offs for encoded distributed optimization

A. Salman Avestimehr*, Seyed Mohammadreza Mousavi Kalan* and Mahdi Soltanolkotabi*,†
Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, USA
Emails: avestimehr@ee.usc.edu mmousavi@usc.edu
†Corresponding author. Email: soltanol@usc.edu

[Received on 25 October 2018; revised on 14 July 2020; accepted on 1 September 2020]

Dealing with the shear size and complexity of today's massive data sets requires computational platforms that can analyze data in a parallelized and distributed fashion. A major bottleneck that arises in such modern distributed computing environments is that some of the worker nodes may run slow. These nodes a.k.a. stragglers can significantly slow down computation as the slowest node may dictate the overall computational time. A recent computational framework, called encoded optimization, creates redundancy in the data to mitigate the effect of stragglers. In this paper, we develop novel mathematical understanding for this framework demonstrating its effectiveness in much broader settings than was previously understood. We also analyze the convergence behavior of iterative encoded optimization algorithms, allowing us to characterize fundamental trade-offs between convergence rate, size of data set, accuracy, computational load (or data redundancy) and straggler toleration in this framework.

Keywords: distributed computing; machine learning; optimization; stragglers.

1. Introduction

Modern data sets are massive in size and complexity consisting of tens of billions of examples. These data sets are also very high-dimensional with numerous detailed information gathered for each example. Furthermore, due to the proliferation of a variety of personal devices many modern data sets are stored or collected in a distributed manner. To process such data sets in a timely manner, distributed computing algorithms/platforms that can analyze data in a parallelized or fully decentralized fashion are crucial.

As we scale out computations across many distributed nodes in modern distributed computing environments, such as Amazon EC2, a major performance bottleneck is the latency in waiting for slowest nodes, or 'stragglers' to finish their tasks [3]. These stragglers are caused by various forms of 'system noise' (e.g. deallocation of computational resources, bandwidth limitation, node failure, etc.) and can significantly slow down computation as the slowest node may dictate the overall computational time. The conventional approaches to mitigate the impact of stragglers involve creation of some form of 'computational redundancy'. For example, *replicating* the straggling task on another available node is a common approach to deal with stragglers (e.g. [23]).

Asynchronous methods like [19] can partially resolve the effect of stragglers but because of system randomness, they may not be reproducible or consistent. More recent approaches [4,10–13,22] bring to bear ideas from coding theory to distributed computing in a synchronous setting. These *coded computing*

* Authors ordered alphabetically.

approaches create redundancy in the *computation tasks* in unorthodox coded forms (as opposed to conventional replication approaches), thereby alleviating the effect of stragglers more efficiently. In the literature of distributed coded computing, there are mainly two approaches. The goal is either to exactly recover the computations done by the workers or just approximate the computations at the master node. For instance, [6,20] propose a coding scheme to exactly recover the gradient sums in the presence of stragglers at the master while the goal of [18] is to approximate the sum of gradients. Charles *et al*. [2] aims at approximately recovering a sum of functions using sparse random graphs and [21] analyzes the convergence and delay properties of gradient descent in the setting of approximate recovery.

Coded computing has also been proposed for creating redundancy in distributed optimization problems [7,8,24,25]. Zhu *et al*. [25] proposes a sequential framework for solving optimization problems using a distributed platform that solves a sequence of optimization problems in place of the original problem. Zhang *et al*. [24] proposes differential coded compressors used in network distributed optimization that has high convergence speed and also relaxes the assumption of bounded noise power on compressors. The key idea of [7,8], named *encoded optimization*, is to linearly encode the data variables in the optimization. The encoded data is then distributed across the computational nodes and distributed optimization algorithms are then applied to these encoded data. Due to the redundancy created in the data, the optimization algorithms can be completed without having to wait for the straggler nodes.

The encoded optimization framework provides an intriguing approach to deal with the effect of stragglers. However, our mathematical understanding of the effect of this data encoding strategy is limited. In particular, existing results such as [7,8] mostly focus on understanding the effect of random encoding strategies on the optimal solution to unconstrained least-squares problems. Furthermore, there is very limited understanding of how such encoding strategies affect the use of various computational and data resources. This is particularly important as in many modern applications ranging from imaging to online advertisement and financial trading we are interested in algorithms that can operate under multiple constraints (e.g. under a limited time budget). Efficient learning from encoded data under these constraints poses new challenges: how can we incorporate domain-specific prior knowledge in a principled manner? What algorithms should we use under a fixed time budget? How much of the data should we use? Should we use all of the data or just parts of it? How many passes (or iterations) of the algorithm is required to get to an accurate solution? How much redundancy should we create in our data? How does the amount of redundancy present in our data encoding strategy affect the convergence behavior and run-time of our algorithms? How many straggler nodes can a particular form of data encoding approach tolerate?

At the heart of answering these questions is the ability to predict run-time of encoded optimization algorithms as a function of the required accuracy, the size of data, the number of straggler nodes, the amount of prior knowledge, etc. That is, we need to understand precise trade-offs between run time, data size, accuracy, data redundancy and straggler toleration of iterative encoded optimization algorithms. In this paper, we wish to precisely characterize such trade-offs, significantly broadening our current understanding of the encoded optimization paradigm. Our main contributions in this paper are as follows.

- We study the encoded optimization framework in a much broader setting than previously understood. In particular, we demonstrate how prior knowledge can be incorporated in this framework via constraints on the optimization variables. Our guarantees are very general and can deal with arbitrary and potentially non-convex constraints.
- Our results require a near minimal amount of data redundancy/replication (a.k.a. computational load). We show that encoded optimization is effective as long as the data redundancy/replication exceeds (up to constants) the sum of the total number of stragglers and a precise quantity capturing

the amount of prior knowledge that is enforced in the optimization algorithm. In fact, in certain cases our framework applies even when the number of encoded data is less than the number of data points allowing for data compression in lieu of redundancy/replication.

- We also precisely characterize the convergence rate of iterative encoded optimization algorithms as a function of various parameters including the straggler toleration, computational load, prior knowledge, as well as the size of the data set. This allows us to precisely characterize the various trade-offs between these fundamental resources.

2. Problem formulation

In this section, we discuss the encoded distributed optimization framework and formulate the underlying fundamental resource trade-offs that we study in this paper.

2.1 *Setting*

In many modern applications in signal processing and machine learning, we aim to infer models that best explain the training data. Given training data consisting of n pairs of input features $\boldsymbol{x}_i \in \mathbb{R}^d$ and desired outputs $y_i \in \mathbb{R}$, we wish to infer a function that best explains the training data.The simplest functions are linear ones where the outputs are linear functions of the features. Specifically, we are interested in finding a parameter $\boldsymbol{\theta}^* \in \mathbb{R}^d$ obeying the following equations:

$$y_i =< \boldsymbol{x}_i, \boldsymbol{\theta}^* > + \boldsymbol{w}_i \quad \text{for} \quad i = 1, 2, ..., n. \tag{2.1}$$

Here, w_i denotes the noise present in our training examples. A natural approach to finding the best linear model is to minimize the empirical risk $\sum_{k=1}^{n} \ell(\langle \boldsymbol{x}_k, \boldsymbol{\theta} \rangle, y_k)$ via a quadratic loss $\ell(u, v) = \frac{1}{2}(u-v)^2$. This leads to the following optimization problem:

$$\begin{aligned} \hat{\boldsymbol{\theta}} &= \arg\min_{\boldsymbol{\theta}} \mathscr{L}(\boldsymbol{\theta}) := \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|_{\ell_2}^2 \\ &\text{subject to} \quad \mathscr{R}(\boldsymbol{\theta}) \leqslant R. \end{aligned} \tag{2.2}$$

Here, $\boldsymbol{y} \in \mathbb{R}^n$ is the output vector consisting of the outputs $\boldsymbol{y} = [y_1\ y_2\ ...\ y_n]^T$, and $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ is the feature matrix consisting of the data features $\boldsymbol{X} = \left[\boldsymbol{x}_1^T\ \boldsymbol{x}_2^T\ ...\ \boldsymbol{x}_n^T\right]^T$. Also, $\mathscr{R} : \mathbb{R}^d \to \mathbb{R}$ is a regularizer function that is used to avoid over-fitting and captures some notion of structure/complexity of the unknown parameter (with R a tuning parameter). We note that while we will focus on linear models and quadratic losses, many of the algorithms and technical proofs in this paper generalize to other models/losses. We aim to pursue these extensions in future publications.

To solve optimization problems of the form (2.2) involving massive data sizes, we need to utilize modern distributed computing platforms. While there are many popular distributed computing schemes [**17**], in this paper we focus on a OneToAll scheme that consists of a master node and L workers. We focus on a distributed implementation of projected gradient descent (PGD) for solving problems of the form (2.2). To distribute PGD, we assume the master partitions the features matrix and the response vector $\boldsymbol{y}$ across rows between L worker nodes. Specifically, we partition $\boldsymbol{X}/\boldsymbol{y}$ into L parts across rows with $\boldsymbol{X} = \left[\boldsymbol{X}_1^T\ \ \boldsymbol{X}_2^T\ \ \dots\ \ \boldsymbol{X}_L^T\right]^T, \boldsymbol{y} = \left[\boldsymbol{y}_1^T\ \ \boldsymbol{y}_2^T\ \ \dots\ \ \boldsymbol{y}_L^T\right]^T$. Here, $\boldsymbol{X}_\ell \in \mathbb{R}^{n_\ell \times d}$ and $\boldsymbol{y}_\ell \in \mathbb{R}^{n_\ell}$ with $\sum_\ell n_\ell = n$.

With this partition, worker ℓ receives data $\boldsymbol{X}_\ell/\boldsymbol{y}_\ell$ and carries out updates/computations on this data. Starting from some initial solution $\boldsymbol{\theta}_0 \in \mathbb{R}^d$, in each iterations the master sends the current update $\boldsymbol{\theta}_\tau$ to the workers. Each of the workers then calculates a partial gradient based on the portion of the data it has access to. Specifically, worker ℓ returns to the master the following partial gradient:

$$\nabla \mathscr{L}^{(\ell)}(\boldsymbol{\theta_\tau}) = \boldsymbol{X}_\ell^T \left(\boldsymbol{X}_\ell \boldsymbol{\theta}_\tau - \boldsymbol{y}_\ell\right).$$

The master then aggregates all of these partial gradients and performs the following update:

$$\begin{aligned}\boldsymbol{\theta}_{\tau+1} &= \mathscr{P}\left(\boldsymbol{\theta}_\tau - \tilde{\mu}_\tau \sum_{\ell=1}^{L} \nabla \mathscr{L}^{(\ell)}(\boldsymbol{\theta}_\tau)\right) \\ &= \mathscr{P}\left(\boldsymbol{\theta}_\tau - \tilde{\mu}_\tau \sum_{\ell=1}^{L} \boldsymbol{X}_\ell^T \left(\boldsymbol{X}_\ell \boldsymbol{\theta}_\tau - \boldsymbol{y}_\ell\right)\right) \\ &= \mathscr{P}\left(\boldsymbol{\theta}_\tau - \tilde{\mu}_\tau \boldsymbol{X}^T (\boldsymbol{X}\boldsymbol{\theta}_\tau - \boldsymbol{y})\right). \end{aligned} \tag{2.3}$$

Here, $\mathscr{P}$ denotes the Euclidean projection onto the constraint set $\mathscr{K} = \{\boldsymbol{\theta} \in \mathbb{R}^d : \mathscr{R}(\boldsymbol{\theta}) \leqslant R\}$ and $\tilde{\mu}_\tau$ is the learning rate.

As mentioned in Section 1, a major performance bottleneck that arises when implementing such distributed PGD updates is that some of the worker nodes may run slow (i.e. stragglers). The presence of such stragglers can significantly slow down the computations as the master has to wait for all the workers to send their partial gradient calculations, so that the overall run time is limited by the slowest worker. For instance, in [1] the effect of slow workers under the title of *outliers* was studied and it was shown that completion time of jobs can be prolonged by 34% at median. In this paper, we will focus on the *encoded optimization framework*, which will be described next, to mitigate the effect of stragglers.

2.2 *The encoded optimization framework*

To deal with the effect of stragglers in the iterations (2.3), in this paper we utilize a new approach for straggler mitigation, named *encoded distributed optimization* [7,8] , which was originally developed for unconstrained least-squares problems. The main idea behind this approach is to create redundancy in the data by random embedding/encoding. In this section, we discuss this computational paradigm tailored to distributed PGD iterates.

To overcome the computational slowdown caused by stragglers, we randomly embed/encode the data by multiplying the feature matrix and the response vector by an encoding matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. We then partition these embedded data $\boldsymbol{AX}/\boldsymbol{Ay}$ and then distribute them across rows between the L worker nodes. Specifically, we partition the matrix $\boldsymbol{A}$ into L parts across rows $\boldsymbol{A} = \begin{bmatrix}\boldsymbol{A}_1^T & \boldsymbol{A}_2^T & \dots & \boldsymbol{A}_L^T\end{bmatrix}^T$ with $\boldsymbol{A}_\ell \in \mathbb{R}^{n_\ell \times n}$ and $m = \sum_\ell n_\ell$. Similar to the un-coded case, with this partition worker ℓ receives data $\boldsymbol{A}_\ell \boldsymbol{X}/\boldsymbol{A}_\ell \boldsymbol{y}$ and carries out updates/computations on this data so that the partial gradient updates are now calculated based on these randomly encoded data. That is,

$$\nabla \mathscr{L}^{(\ell)}(\boldsymbol{\theta_\tau}) = \boldsymbol{X}^T \boldsymbol{A}_\ell^T \boldsymbol{A}_\ell \left(\boldsymbol{X}\boldsymbol{\theta}_\tau - \boldsymbol{y}\right).$$

Due to the effect of stragglers, the master may not receive gradient updates from some workers in a timely manner so that computations based on some of the m rows maybe missing. Let us denote the index of the slow workers at iteration τ by $\mathscr{I}_\tau \subset \{1, 2, \ldots, L\}$. Also let $\mathscr{W}_\ell \subset \{1, 2, \ldots, m\}$ denote the index of the rows of $\boldsymbol{A}$ sent to worker ℓ. Now define

$$\mathscr{S}_\tau = \bigcup_{\ell \in \mathscr{I}_\tau} \mathscr{W}_\ell,$$

which contains the indices of all the rows of $\boldsymbol{A}$ that is not available at the master due to stragglers. We will use $s_\tau = \left|\mathscr{S}_\tau\right|$ to denote the total number of these straggler rows.

Based on the gradient updates available to the master, it proceeds with the following PGD update:

$$\begin{aligned}\boldsymbol{\theta}_{\tau+1} &= \mathscr{P}\left(\boldsymbol{\theta}_\tau - \mu_\tau \sum_{\ell \in \mathscr{S}_\tau^c} \nabla \mathscr{L}^{(\ell)}(\boldsymbol{\theta}_\tau)\right) \\ &= \mathscr{P}\left(\boldsymbol{\theta}_\tau - \mu_\tau \boldsymbol{X}^T \boldsymbol{A}_{\mathscr{S}_\tau^c}^T \boldsymbol{A}_{\mathscr{S}_\tau^c} (\boldsymbol{X}\boldsymbol{\theta}_\tau - \boldsymbol{y})\right), \end{aligned} \tag{2.4}$$

where μ_τ is the learning rate of encoded iterations. Therefore, the master effectively runs the encoded iterations (2.4) in lieu of the uncoded iterates (2.3).

We would like to note that to compute the coded data $\boldsymbol{A}_\ell \boldsymbol{X}/\boldsymbol{A}_\ell \boldsymbol{y}$, the distributed nodes do not need access to the full data. In many applications, it is common for the master to divide the data set between the worker nodes. Thus, the computation $\boldsymbol{A}_\ell \boldsymbol{X}/\boldsymbol{A}_\ell \boldsymbol{y}$ can be carried out once at the master and then coded batches are distributed between the worker nodes. Moreover, if the size of data is more than that the master be capable of computing the coded data they can be computed over a cloud and then distributed among the workers. We also note that while encoding the data (i.e. $\boldsymbol{A}\boldsymbol{X}/\boldsymbol{A}\boldsymbol{y}$) maybe costly this is a 'one-time cost'. In many cases, this is negligible compared to the computational cost of training various models which require many iterations. Furthermore, once coded the data can be used to train more than one model. That said, while our theory currently focuses on Gaussian encoding matrices, we intend to extend our results to coding matrices that admit fast matrix-vector multiplication such as those involving randomized Fourier or sparse matrices. In fact, we demonstrate the effectiveness of such matrices in our numerical experiments.

Note that, apriori it is not clear when/why the encoded iterates serve as a good proxy for the uncoded ones. Understanding this relationship is the main focus of this paper. In the next section, we discuss the main problems that we study in this paper by formalizing various fundamental trade-offs that arise in the distributed encoded optimization framework.

2.3 *Fundamental resource trade-offs*

In this paper, we wish to understand under what conditions the encoded iterates (2.4) are a good proxy for the uncoded iterates (2.3). We aim to answer fundamental questions such as: when will both set of iterates converge to the same fixed point? How does the convergence behavior change due to the presence of the encoding mapping? What are the various trade-offs involved between various resources. To discuss these problems more precisely, we start with two simple definitions related to the encoding matrix $\boldsymbol{A}$.

DEFINITION 2.1 (Computational load). We use computational load m to refer to the number of rows of $\boldsymbol{A}$.

DEFINITION 2.2 (Straggler toleration). We use s to denote the maximum number of straggler rows in each iteration ($\left|\mathscr{S}_\tau\right| \leqslant s$). We refer to this quantity as straggler toleration.

With these definitions in place, we now discuss the fundamental trade-offs that we aim to characterize in this paper.

- **Accuracy vs. computational time.** In many modern learning applications, we must operate on a fixed time budget. Therefore, it is crucial to understand how many passes (or iterations) of the algorithm is required to get to a certain accuracy. We wish to characterize this fundamental trade-off between computational time and accuracy for the encoded distributed optimization framework. Stated more formally, we are interested in precisely understanding the distance between the encoded iterates and the true parameter ($\|\boldsymbol{\theta}_\tau - \boldsymbol{\theta}^*\|_{\ell_2}$) as a function of the number of iterations (τ) and the noise level $\|\boldsymbol{w}\|_{\ell_2}$.
- **Convergence rate vs. computational load.** We are interested in understanding how the computational load m affects the convergence behavior of the encoded iterates. Intuitively, as the computational load increases the encoded iterates provide a better approximation to the uncoded iterates. Therefore, we expect the coded iterates to converge faster as the computational load increases. We wish to precisely characterize the convergence rate as a function of the computational load.
- **Convergence rate vs. straggler toleration.** In each encoded iteration, there are some stragglers that are ignored by the master node. We aim to characterize the impact of stragglers on the speed of convergence. By increasing the number of stragglers (s), the master node ignores more and more data. Therefore, intuitively we expect that the more stragglers we have, the more iterations are needed for the encoded iterates to converge to a certain accuracy. We wish to characterize the convergence rate as a function of the straggler toleration parameter.
- **Computational load vs. straggler toleration.** Intuitively, as we increase the number of stragglers, s, we need more redundancy in our encoded framework. Stated differently, we need to increase the computational load as a function of the number of stragglers. This leads to a fundamental trade-off between computational load and straggler toleration. We aim to characterize the minimum required computational load as a function of the straggler toleration parameter so as to ensure the encoded iterates eventually converge to a good estimate.

In the next section, we state our main result that leads to a precise characterization of the convergence behavior of the encoded iterates as a function of various parameters, allowing us to precisely characterize the above trade-offs.

3. Main results

We wish to characterize the convergence behavior of the encoded iterates (2.3) as a function of various problem parameters for the worse possible choice of s straggler rows. More precisely, we are interested in characterizing the relationship between

$$\sup_{\mathscr{S}_\tau \subset \{1,2,\ldots,m\},\ \left|\mathscr{S}_\tau\right| \leqslant s} \left\|\boldsymbol{\theta}_\tau - \boldsymbol{\theta}^*\right\|_{\ell_2},$$

and the error $\|\boldsymbol{w}\|_{\ell_2}$ when running the iterations (2.3). To make these connections precise and quantitative, we need a few definitions.

Naturally, our results depend on how well the regularization function $\mathscr{R}$ can capture the properties of the unknown parameter $\boldsymbol{\theta}^*$. For example, if we know our unknown parameter is approximately sparse, then using an ℓ_1 norm for the regularizer is superior to using an ℓ_2 regularizer. To quantify this capability, we first need a couple of standard definitions which we adapt from [14, 15].

DEFINITION 3.1 (Descent set and cone). The *set of descent* of a function $\mathscr{R}$ at a point $\boldsymbol{\theta}^*$ is defined as

$$D_{\mathscr{R}}(\boldsymbol{\theta}^*) = \left\{\boldsymbol{h} : \ \mathscr{R}(\boldsymbol{\theta}^* + \boldsymbol{h}) \leqslant \mathscr{R}(\boldsymbol{\theta}^*)\right\}.$$

The *cone of descent* is defined as a closed cone $\mathscr{C}_{\mathscr{R}}(\boldsymbol{\theta}^*)$ that contains the descent set, i.e. $\mathscr{D}_{\mathscr{R}}(\boldsymbol{\theta}^*) \subset \mathscr{C}_{\mathscr{R}}(\boldsymbol{\theta}^*)$. The *tangent cone* is the conic hull of the descent set. That is, the smallest closed cone $\mathscr{C}_{\mathscr{R}}(\boldsymbol{\theta}^*)$ obeying $\mathscr{D}_{\mathscr{R}}(\boldsymbol{\theta}^*) \subset \mathscr{C}_{\mathscr{R}}(\boldsymbol{\theta}^*)$.

We note that the capability of the regularizer $\mathscr{R}$ in capturing the properties of the parameter vector $\boldsymbol{\theta}^*$ depends on the size of the descent cone $\mathscr{C}_{\mathscr{R}}(\boldsymbol{\theta}^*)$. The smaller this cone is the more suited the function $\mathscr{R}$ is at capturing the properties of $\boldsymbol{\theta}^*$. To quantify the size of various cones, we shall use the notion of mean width.

DEFINITION 3.2 (Gaussian width). The Gaussian width of a set $\mathscr{C} \in \mathbb{R}^n$ is defined as $\omega(\mathscr{C}) := \mathbb{E}_{\boldsymbol{g}}[\sup_{\boldsymbol{z}\in\mathscr{C}}\langle \boldsymbol{g}, \boldsymbol{z}\rangle]$, where the expectation is taken over $\boldsymbol{g} \sim \mathscr{N}(\boldsymbol{0}, \boldsymbol{I}_n)$.

We now have all the definitions in place to quantify the capability of the function $\mathscr{R}$ in capturing the properties of the unknown parameter $\boldsymbol{\theta}^*$ when using an encoding matrix $\boldsymbol{A}$. This naturally leads us to the definition of the minimum required computational load.

DEFINITION 3.3 (Minimal computational load). Let $\mathscr{C}_{\mathscr{R}}(\boldsymbol{\theta}^*)$ be a cone of descent of $\mathscr{R}$ at $\boldsymbol{\theta}^*$. We define the minimal computational load function as

$$\mathscr{M}(\mathscr{R}, \boldsymbol{X}, \boldsymbol{\theta}^*, \eta) = \left(\omega\left(\boldsymbol{X}\mathscr{C}_{\mathscr{R}}(\boldsymbol{\theta}^*) \cap \mathbb{S}^{n-1}\right) + \eta\right)^2,$$

where $\boldsymbol{\theta}^* \in \mathbb{R}^d$, $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ and the variable $\eta \in \mathbb{R}$. We shall often use the short hand $m_0 = \mathscr{M}(\mathscr{R}, \boldsymbol{X}, \boldsymbol{\theta}^*, \eta)$ with the dependence on $\mathscr{R}, \boldsymbol{X}, \boldsymbol{\theta}^*, \eta$ implied.

The definition above characterizes the minimum computational load required for the encoded iterations to converge to the true parameter in the absence of noise or stragglers.

The convergence rate of the encoded iterates also naturally depends on various characteristics of the feature matrix $\boldsymbol{X}$. We quantify a few of these characteristics below.

DEFINITION 3.4 (Cone-restricted spectral norm). Let $\mathscr{C}_{\mathscr{R}}(\boldsymbol{\theta}^*)$ be the cone of descent of the regularization function $\mathscr{R}$ at a point $\boldsymbol{\theta}^*$ per Definition 3.1. The cone-restricted spectral norm of a matrix $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ with respect to $\mathscr{R}$ at a point $\boldsymbol{\theta}^*$ is defined as $\sigma_{\mathscr{R}}(\boldsymbol{X}) = \sup_{\boldsymbol{u}\in\mathscr{C}_{\mathscr{R}}(\boldsymbol{\theta}^*)\cap\mathbb{S}^{d-1}} \|\boldsymbol{X}\boldsymbol{u}\|_{\ell_2}$.

We note that the above definition is a natural extension of the spectral norm of a matrix. It is well known that the spectral norm of the feature matrix plays a crucial role in the convergence behavior of least-square problems. The cone-restricted spectral norm defined above plays a similar role in the convergence of constrained least-squares problems.

Furthermore, the convergence behavior of the encoded iterates is also related to that of the uncoded iterates. The following two definitions, adapted from [14], concern the convergence of the uncoded iterates.

DEFINITION 3.5 (Convergence rate). Consider the iterations 2.3. Let $\boldsymbol{\theta}^* \in \mathbb{R}^d$ and $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ and $\mathscr{R}$ be the regularizer function as well as μ be the learning rate. We define

$$\rho(\mu) := \rho(\mu, \boldsymbol{X}, \mathscr{R}, \boldsymbol{\theta}^*) = \sup_{\boldsymbol{u},\boldsymbol{v}\in\mathscr{C}_{\mathscr{R}}(\boldsymbol{\theta}^*)\cap\mathbb{S}^{d-1}} \boldsymbol{u}^T\left(\boldsymbol{I} - \mu\boldsymbol{X}^T\boldsymbol{X}\right)\boldsymbol{v}.$$

It is known that $\rho(\mu)$ characterizes the convergence rate of the uncoded iterations (2.3) [14].

DEFINITION 3.6 (Noise amplification). Consider the iterations 2.3. Let $\boldsymbol{\theta}^* \in \mathbb{R}^d$ and $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ and $\mathscr{R}$ be the regularizer function with $\boldsymbol{w}$ denoting the noise. We define

$$\xi(\boldsymbol{X}) := \xi(\boldsymbol{X}, \mathscr{R}, \boldsymbol{\theta}^*, \boldsymbol{w}) = \sup_{\boldsymbol{v}\in-\mathscr{C}_{\mathscr{R}}(\boldsymbol{\theta}^*)\cap\mathbb{S}^{d-1}} \boldsymbol{v}^T\boldsymbol{X}^T\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_{\ell_2}}.$$

It is known that the uncoded iterations eventually converge to a neighborhood of the unknown parameter $\boldsymbol{\theta}^*$ [14]. The noise amplification factor defined above plays a crucial role in characterizing the size of this neighborhood. In particular, [14] shows that the diameter of this neighborhood is proportional to $\xi(\boldsymbol{X})\,\|\boldsymbol{w}\|_{\ell_2}$.

With these definitions in place, we are now ready to state our main theorem regarding the convergence of the encoded iterates (2.4).

THEOREM 3.7 Let $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ be a matrix with i.i.d. $\mathscr{N}(0,1)$ entries. Also, assume the number of straggler rows obeys $s_\tau = |\mathscr{S}_\tau| \leqslant s$ with s the straggler toleration parameter per Definition 2.2 obeying $s \leqslant m$. Furthermore, let $m_0 = \mathscr{M}(\mathscr{R}, \boldsymbol{X}, \boldsymbol{\theta}^*, \eta)$ denote the minimal computational load per Definition 3.3. Then the encoded iterative updates (2.4) obey

$$\begin{aligned}
&\sup_{\mathscr{S}_\tau\subset\{1,2,\ldots,m\}} \left\|\boldsymbol{\theta}_{\tau+1} - \boldsymbol{\theta}^*\right\|_{\ell_2} \\
&\quad\leqslant \kappa_{\mathscr{R}}\rho(\tilde{\mu}_\tau)\left\|\boldsymbol{\theta}_\tau - \boldsymbol{\theta}^*\right\|_{\ell_2} + \tilde{\mu}_\tau\cdot\kappa_{\mathscr{R}}\cdot\sigma^2_{\mathscr{R}}(\boldsymbol{X})\cdot\left(\frac{2+9s_\tau\log(em/s_\tau)}{m} + 4\sqrt{\frac{m_0}{m-s_\tau}}\right)\left\|\boldsymbol{\theta}_\tau - \boldsymbol{\theta}^*\right\|_{\ell_2} \\
&\quad\quad + \kappa_{\mathscr{R}}\left(\tilde{\mu}_\tau\cdot\xi(\boldsymbol{X}) + \frac{\tilde{\mu}_\tau}{\sqrt{2}}\cdot\sigma_{\mathscr{R}}(\boldsymbol{X})\sqrt{\frac{m_0}{m-s_\tau}}\right)\|\boldsymbol{w}\|_{\ell_2}, \qquad (3.1)
\end{aligned}$$

for all τ with probability at least $1 - 6e^{-\frac{\eta^2}{8}} - e^{-\frac{\eta^2}{2}} - e^{-\frac{m}{2}}$. Here, ρ is the convergence rate per Definition 3.5, ξ is the noise amplifications per Definition 3.6, $\sigma_{\mathscr{R}}(\boldsymbol{X})$ is the cone-restricted spectral norm of $\boldsymbol{X}$ per Definition 3.4. Furthermore, the tuning parameter is set to $R = \mathscr{R}(\boldsymbol{\theta}^*)$ and the learning rate is equal to $\mu_\tau = \frac{\tilde{\mu}_\tau}{\beta^2_{s_\tau,m}}$ with $\beta_{s,m} = \min\left(\sqrt{3(m-s)\log\left(\frac{em}{m-s}\right)}, \sqrt{m}\right)$ and $\tilde{\mu}_\tau$ is the learning rate in the uncoded iterations (2.3). Finally, $\kappa_{\mathscr{R}} = 1$ for convex $\mathscr{R}$ and $\kappa_{\mathscr{R}} = 2$ for non-convex $\mathscr{R}$.

REMARK 3.8 Theorem 3.7 essentially connects the convergence behavior of the encoded iterations to that of the uncoded iterations. Consider the limit $m \to \infty$ and note that for a Gaussian matrix $\boldsymbol{A}_{\mathscr{S}_\tau^c}$,

$\boldsymbol{A}_{\mathscr{S}_\tau}^T \boldsymbol{A}_{\mathscr{S}_\tau} \to (m - s_\tau)\boldsymbol{I}$ and thus the encoded iterations reduce to the uncoded iterations (modulo a constant factor in the step size). In this case, the convergence bound provided by Theorem 3.7 reduces to

$$\left\|\boldsymbol{\theta}_{\tau+1} - \boldsymbol{\theta}^*\right\|_{\ell_2} \leqslant \kappa_{\mathscr{R}} \rho(\tilde{\mu}_\tau) \left\|\boldsymbol{\theta}_\tau - \boldsymbol{\theta}^*\right\|_{\ell_2} + \frac{\kappa_{\mathscr{R}}}{\sqrt{2}} \tilde{\mu}_\tau \cdot \xi(\boldsymbol{X})) \left\|\boldsymbol{w}\right\|_{\ell_2}. \tag{3.2}$$

The first term gives the convergence rate to the true parameter. The second term characterizes the size of the neighborhood (of the true parameter) to which the iterates converge, demonstrating that the iterates eventually approximate the true parameter up to a term that is proportional to the Euclidean norm of the noise. The bound (3.2) was proven recently in [**14**] . Our result generalizes this result to the encoded case while recovering the special uncoded case in the limit $m \to \infty$.

REMARK 3.9 Theorem 3.7 characterizes the minimal computational load for convergence of (2.4) in the presence of stragglers. Comparing (3.1) with (3.2), we see that as long as the computational load is sufficiently large the effect of coding is only a slight increase in the convergence rate and the size of approximation neighborhood. For instance, to ensure that in the encoded case the convergence rate only increases by ϵ the computational load must obey

$$m - s \geqslant 260 \frac{m_0 + s}{\epsilon^2} \geqslant 64 \frac{m_0 + s}{\epsilon^2} \kappa_{\mathscr{R}}^2 \big(\tilde{\mu}_\tau \cdot \sigma_{\mathscr{R}}^2(\boldsymbol{X})\big)^2.$$

In the last inequality, we used $\kappa_{\mathscr{R}} \leqslant 2$ and the fact that $\tilde{\mu}_\tau$ typically scales with $1/\sigma_{\mathscr{R}}(\boldsymbol{X})^2$ (as step size typically scales with the inverse of the smoothness parameter). Thus, as long as the computational load exceeds the sum of the number of stragglers and the minimal computational load by a constant factor, i.e.

$$m \geqslant c(m_0 + s) \tag{3.3}$$

holds for some numerical constant c depending only on ϵ, then the increase in the convergence rate is small. Similarly, the increase in the size of the approximation neighborhood remains small as long as (3.3) holds.

REMARK 3.10 We now briefly discuss how our results compare with related work. Theorem 3.7 demonstrates that the encoded iterates converge at a linear rate while dealing with arbitrary and possibly non-convex constraints. Karakus *et al.* [**8**] also demonstrates a linear convergence, albeit in terms of the optimal value. However, [**8**] only focuses on the special case where there are no constraints on the optimization variables. Furthermore, [**8**] requires a computational load that is larger than the sum of the number of stragglers and the total number of data points, i.e. $m \geqslant n + 2s$. In comparison, our results require a near minimal number of samples that is commensurate to the sum of the straggler toleration and the amount of prior knowledge ($m \geqslant c(m_0 + s)$). This allows for a much smaller computation load that can even be significantly smaller than the number of data points, i.e. $m << n$. Finally, we would like to mention related work in [**16**] where the authors focus on sketching of constrained convex programs. This paper focuses on the properties of the optimal solution to problems of the form (2.2) without any stragglers. In comparison, we focus on analyzing the convergence behavior of iterative algorithms when stragglers are present.

REMARK 3.11 Theorem 3.7 focuses on an adversarial model for stragglers. Specifically, when bounding the error in each iteration we assume a maximal number and adversarial form/location of stragglers in

each iteration. However, we can apply this theorem for other models such as random faults in which number/location of stragglers varies in each iteration in a random fashion. Furthermore, we assume that every straggler is a full straggler which means that it completely fails and cannot send any result to the master. It may be the case that some stragglers fail completely but some others are slow compared to normal workers but can still send some results to the master. We believe that our results can be extended to deal with these scenarios and hope to address this in future work.

REMARK 3.12 In Theorem 3.7, we assume that the parameter R is tuned perfectly and is set to $R = \mathscr{R}(\boldsymbol{\theta}^*)$. It is not necessary to know $\mathscr{R}(\boldsymbol{\theta}^*)$ in advance. For uncoded iterations and in the absence of stragglers [**14**, Theorem 2.6] discusses the effect of not setting R to $R = \mathscr{R}(\boldsymbol{\theta}^*)$ and characterizes the effect of this mismatch. Theorem 3.7 can also be generalized in a similar fashion to account for this mismatch.

REMARK 3.13 (Convergence rate vs. computational load). Theorem 3.7 characterizes the effect of the computational load on the convergence rate. In particular, this theorem shows that the increase in the convergence rate is proportional to $1/\sqrt{m}$. Therefore, as the computational load increases the convergence rate decreases. Thus, a larger computational load ensures a faster convergence of the encoded iterates.

REMARK 3.14 (Convergence rate vs. straggler toleration). Theorem 3.7 also characterizes the effect of stragglers on the rate of convergence. This result demonstrates a rate proportional to $1/\sqrt{m-s}$ so that as the number of stragglers increase, the convergence rate decreases leading to a slower convergence of the encoded iterates.

REMARK 3.15 (Computational load vs. straggler toleration). We also note that Theorem 3.7 indirectly characterizes a trade-off between the computational load and the straggler toleration of the encoded iterations through (3.3). Indeed, (3.3) demonstrates that for a fixed convergence rate the computational load must scale linearly with the number of stragglers.

4. Numerical results

In this section, we corroborate the resource trade-offs characterized in Theorem 3.7 via experiments on synthetic data. We generate the true parameter $\boldsymbol{\theta}^* \in \mathbb{R}^d$ with $d = 4000$ and sparsity level $k = 20$, where the support is chosen at random and the values on support are distributed i.i.d $N(0,1)$. Moreover, we generate the data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ i.i.d. $\sim N(0,1)$ with $n = 3000$ and set the output vector via $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\theta}^*$. In our simulations, we vary the computational load m and the straggler toleration s and then plot the various trade-offs. We use two different encoding matrices: a random Gaussian matrix and a random discrete cosine transform (DCT) matrix. In the Gaussian case, the entries of the matrix are generated i.i.d. $\sim N(0,1)$. The random DCT matrix is generated according to $\boldsymbol{A} = \boldsymbol{HD}$ where $\boldsymbol{H} \in \mathbb{R}^{m\times n}$ is obtained by selecting m rows of an $n \times n$ DCT matrix at random, and $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with i.i.d. ± 1 entries on the diagonal. In our simulations in each iteration, we assume a different set of straggler rows chosen i.i.d. at random from the m rows. To reconstruct $\boldsymbol{\theta}^*$, we run encoded PGD iterations (2.4) for solving (2.2) with learning rates $\mu_\tau = \frac{1}{5m}$ and $\mu_\tau = 1/3$ for the Gaussian and randomized DCT encoding matrices, respectively. We use $\mathscr{R}(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_{\ell_1}$ with tuning parameter $R = ||\boldsymbol{\theta}^*||_1$. We run the encoded PGD iterates for 500 iterations and record the relative error $\|\boldsymbol{\theta}_\tau - \boldsymbol{\theta}^*\|_{\ell_2}/\|\boldsymbol{\theta}^*\|_{\ell_2}$.

- **Convergence rate vs. computational load.** In this simulation, we fix the straggler toleration at $s = 100$ and vary the computation load m. We depict the relative error as a function of iterations

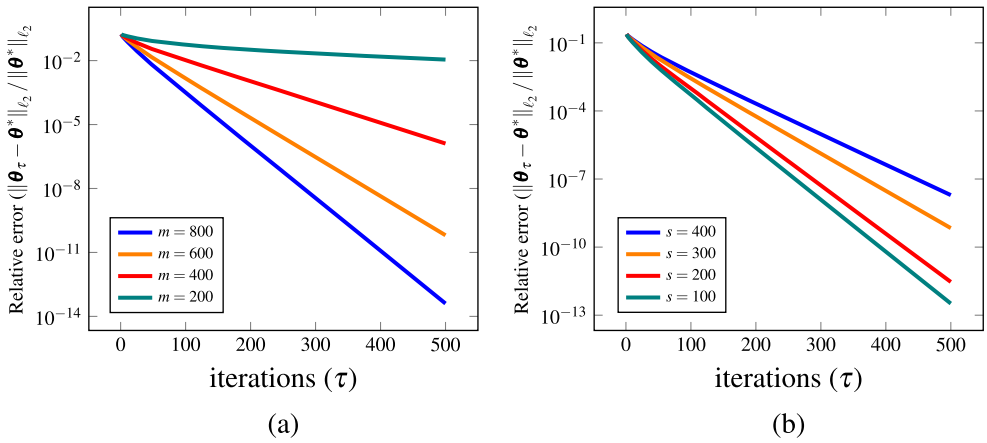

FIG. 1. These two diagram show the empirical rates of convergence for Gaussian encoded PGD in two different scenarios. (a) Depicts the converge rate as a function of the computational load m when the straggler toleration is fixed at $s = 100$. (b) Shows the convergence rate as a function of the straggler toleration with a fixed computational load at $m = 800$.

in Fig. 1(a). This figure confirms that the convergence is indeed linear and increasing m leads to a faster convergence as predicted by Theorem 3.7.

- **Convergence rate vs. straggler toleration.** In this simulation, we fix the computational load at $m = 800$ and vary the straggler toleration s and depict the relative errors as a function of the iterations in Fig. 1(b). This figure confirms that the iterates converge at a linear rate and increasing s leads to a slower convergence as predicted by Theorem 3.7.

- **Computational load vs. straggler toleration.** In this simulation, we vary the computational load m, and straggler toleration s and for each case run the encoded PGD iterations. We stop after 500 iterations and record the empirical probability of success. The empirical probability of success is an average over 50 trials, where in each instance, we generate new random parameter vectors, data and encoding matrices. We declare a trial successful if the relative error of the reconstruction $\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_{\ell_2}/\|\boldsymbol{\theta}^*\|_{\ell_2}$ falls below 10^{-3}.

- Figure 2(a) depicts the empirical success probabilities via a color map for different straggler tolerations s and computational loads m. Yellow represents certain success, while blue represents certain failure. In the experiments of this figure, the encoding matrix is Gaussian. This figure clearly shows that there is a phase transition curve for the computational load as a function of the straggler toleration. On one side of this curve encoded PGD updates is successful with high probability on the other side it fails with high probability. Figure 2(a) also shows that the computational load scales linearly in terms of the straggler toleration parameter confirming the relationship (3.3) predicted by Theorem 3.7. Figure 2(b) depicts the results for randomized DCT matrices. Encoding with such matrices is very efficient requiring only a DCT transform. Perhaps unexpectedly, randomized DCT matrices exhibit very similar behavior to the Gaussian matrix demonstrating that such matrices can act as computational friendly surrogates for encoding purposes. Proving Theorem 3.7 extends to randomized DCT matrices is an interesting direction for future research.

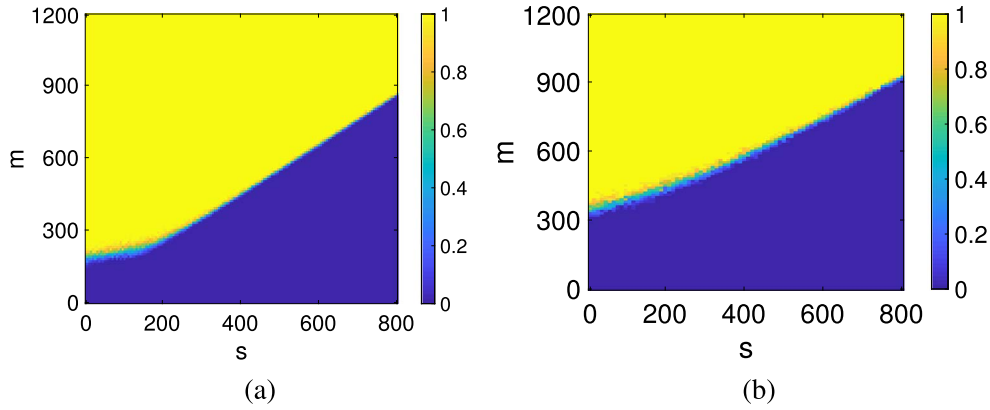

FIG. 2. These two diagrams depict the empirical probability that encoded PGD successfully reaches the global optimum of the uncoded optimization problem for (a) Gaussian and (b) randomized DCT encoding matrices. The colormap tapers between yellow and blue where yellow represents certain success, while blue represents certain failure.

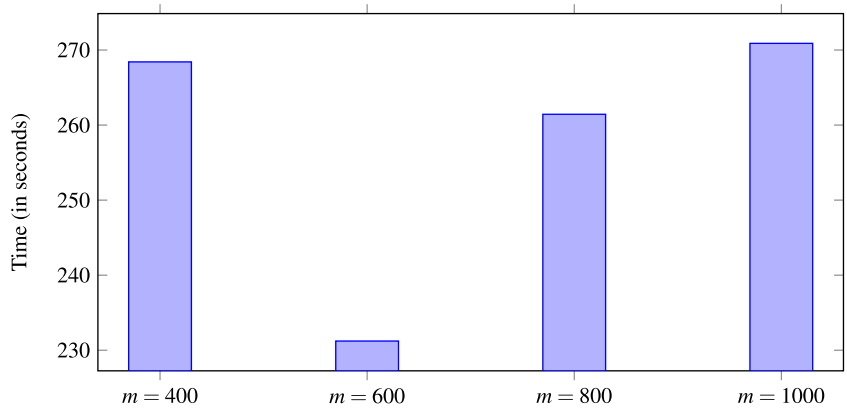

FIG. 3. Illustration of runtime of encoding iterations as a function of m with $s = 100$.

- In the next, we consider another set of experiments in order to investigate the effect of m, size of encoding matrix, on the runtime of iterations. Here we use shifted exponential model proposed in [**10**].
- We consider the same setting described above except that we assume that computing each partial gradient at every iteration takes a random amount of time 10^{-4} + exp(mean) where mean = 10^{-3}. We insert delay and measure the runtime using the commands *pause* as well as *tic, toc* in MATLAB. Furthermore, in various settings, we measure the runtime of reaching the error to 10^{-8}. Figure 3 depicts the effect of m on the runtime of iterations. If we choose too small m then the speed of convergence is low and it takes a huge time to reach to the desired level of error. On the other hand, if we choose too large m the total computation time becomes very large. As Fig. 3 shows there is an optimal m for which the runtime is optimal.

5. Proofs

In this section, we prove Theorem 3.7. In the absence of stragglers, the problem reduces to uncoded iterations. Our proofs is based on relating the coded iterates to uncoded ones that can be seen as the average over the random choice of the coding matrix. Thus, the uncoded iterates can be thought of as a 'population counterpart' and the limit of coded iterations with $m \to \infty$. The proof consists of three main steps. In the first step, we utilize a deterministic convergence analysis from [14, Theorem 1.2] and decompose the convergence rate into two terms. The first term is the population term (averaging over the randomness in the coding matrix) and has the same rate of convergence as the uncoded iterations. The second term is a deviation that captures the perturbation of the convergence rate from its population counterpart. We also perform some simplifications to formulate the deviation in terms of uniform convergence of quadratic stochastic processes. In the second step, we prove a result on the uniform concentration of stochastic processes in the presence of stragglers. In the final step, we combine steps I and II to bound the deviation term and complete the proof.

Step 1: In this step, we decompose the convergence rate into two terms. The first term is the convergence rate of the uncoded iterates and the second term can be thought of the deviation from this expected term. We then cast this deviation in the form of deviations of certain quadratic stochastic processes.

Specifically, define the error vector $\boldsymbol{h}_\tau := \boldsymbol{\theta}_\tau - \boldsymbol{\theta}^*$ and the cones $\widetilde{\mathscr{C}} := \mathscr{C}_{\mathscr{R}}(\boldsymbol{\theta}^*) \in \mathbb{R}^d$ and $\mathscr{C} := \boldsymbol{X}\widetilde{\mathscr{C}} \in \mathbb{R}^n$. Utilizing [14, Theorem 1.2], we have

$$\left\|\boldsymbol{h}_{\tau+1}\right\|_{\ell_2} \leqslant \kappa_{\mathscr{R}}\left(\sup_{\tilde{\boldsymbol{u}},\tilde{\boldsymbol{v}}\in\widetilde{\mathscr{C}}\cap\mathbb{S}^{d-1}} \tilde{\boldsymbol{u}}^T(\boldsymbol{I}-\mu_\tau \boldsymbol{X}^T\boldsymbol{A}_{\mathscr{S}_\tau}^T\boldsymbol{A}_{\mathscr{S}_\tau}\boldsymbol{X})\tilde{\boldsymbol{v}}\right)\left\|\boldsymbol{h}_\tau\right\|_{\ell_2} + \kappa_{\mathscr{R}}\cdot\mu_\tau\cdot \sup_{\tilde{\boldsymbol{u}}\in-\widetilde{\mathscr{C}}\cap\mathbb{S}^{d-1}} \tilde{\boldsymbol{u}}^T\boldsymbol{X}^T\boldsymbol{A}_{\mathscr{S}_\tau}^T\boldsymbol{A}_{\mathscr{S}_\tau}\boldsymbol{w}. \tag{5.1}$$

We now proceed by simplifying each of these two terms. To simplify the first term, define $\boldsymbol{u} := \frac{\boldsymbol{X}\tilde{\boldsymbol{u}}}{\|\boldsymbol{X}\tilde{\boldsymbol{u}}\|_{\ell_2}} \in \mathbb{S}^{n-1}$ and $\boldsymbol{v} := \frac{\boldsymbol{X}\tilde{\boldsymbol{v}}}{\|\boldsymbol{X}\tilde{\boldsymbol{u}}\|_{\ell_2}} \in \mathbb{S}^{n-1}$ and note that

$$\tilde{\boldsymbol{u}}^T\left(\boldsymbol{I}-\mu_\tau\boldsymbol{X}^T\boldsymbol{A}_{\mathscr{S}_\tau}^T\boldsymbol{A}_{\mathscr{S}_\tau}\boldsymbol{X}\right)\tilde{\boldsymbol{v}} = \tilde{\boldsymbol{u}}^T\left(\boldsymbol{I}-\tilde{\mu}_\tau\boldsymbol{X}^T\boldsymbol{X}\right)\tilde{\boldsymbol{v}} + \tilde{\mu}_\tau\cdot\|\boldsymbol{X}\tilde{\boldsymbol{u}}\|_{\ell_2}\cdot\|\boldsymbol{X}\tilde{\boldsymbol{v}}\|_{\ell_2}\,\boldsymbol{u}^T\left(\boldsymbol{I}-\frac{1}{\beta_{s_\tau,m}^2}\boldsymbol{A}_{\mathscr{S}_\tau}^T\boldsymbol{A}_{\mathscr{S}_\tau}\right)\boldsymbol{v}.$$

Now we can use the fact that supremum of sum is less than sum of suprema. Thus,

$$\begin{aligned}
&\sup_{\tilde{\boldsymbol{u}},\tilde{\boldsymbol{v}}\in\widetilde{\mathscr{C}}\cap\mathbb{S}^{d-1}} \tilde{\boldsymbol{u}}^T(\boldsymbol{I}-\mu_\tau\boldsymbol{X}^T\boldsymbol{A}_{\mathscr{S}_\tau}^T\boldsymbol{A}_{\mathscr{S}_\tau}\boldsymbol{X})\tilde{\boldsymbol{v}}\\
&\quad\leqslant \sup_{\tilde{\boldsymbol{u}},\tilde{\boldsymbol{v}}\in\widetilde{\mathscr{C}}\cap\mathbb{S}^{d-1}} \tilde{\boldsymbol{u}}^T\left(\boldsymbol{I}-\tilde{\mu}_\tau\boldsymbol{X}^T\boldsymbol{X}\right)\tilde{\boldsymbol{v}}\\
&\qquad+\tilde{\mu}_\tau\left(\sup_{\tilde{\boldsymbol{u}}\in\widetilde{\mathscr{C}}\cap\mathbb{S}^{d-1}}\|\boldsymbol{X}\tilde{\boldsymbol{u}}\|_{\ell_2}\cdot\sup_{\tilde{\boldsymbol{v}}\in\widetilde{\mathscr{C}}\cap\mathbb{S}^{d-1}}\|\boldsymbol{X}\tilde{\boldsymbol{v}}\|_{\ell_2}\cdot\sup_{\boldsymbol{u},\boldsymbol{v}\in\mathscr{C}\cap\mathbb{S}^{n-1}}\boldsymbol{u}^T\left(\boldsymbol{I}-\frac{1}{\beta_{s_\tau,m}^2}\boldsymbol{A}_{\mathscr{S}_\tau}^T\boldsymbol{A}_{\mathscr{S}_\tau}\right)\boldsymbol{v}\right)\\
&\quad=\rho(\tilde{\mu}_\tau)+\tilde{\mu}_\tau\cdot\sigma_{\mathscr{R}}^2(\boldsymbol{X})\sup_{\boldsymbol{u},\boldsymbol{v}\in\mathscr{C}\cap\mathbb{S}^{n-1}}\boldsymbol{u}^T\left(\boldsymbol{I}-\frac{1}{\beta_{s_\tau,m}^2}\boldsymbol{A}_{\mathscr{S}_\tau}^T\boldsymbol{A}_{\mathscr{S}_\tau}\right)\boldsymbol{v}.
\end{aligned}\tag{5.2}$$

In above, we used the cone-restricted spectral norm of $\boldsymbol{X}$ per Definition 3.4. We now focus on simplifying the second term in 5.1. To this aim, for a vector $\boldsymbol{u} \in \mathbb{R}^n$ define $\boldsymbol{u}_\perp = \boldsymbol{u} - \frac{\boldsymbol{u}^T\boldsymbol{w}}{\|\boldsymbol{w}\|_{\ell_2}^2}\boldsymbol{w}$. We can use this definition to separate the second term in (5.1) into two terms as follows:

$$\begin{aligned}\mu_\tau \cdot \tilde{\boldsymbol{u}}^T\boldsymbol{X}^T\boldsymbol{A}_{\mathscr{S}_\tau^c}^T\boldsymbol{A}_{\mathscr{S}_\tau^c}\boldsymbol{w} &= \mu_\tau \|\boldsymbol{X}\tilde{\boldsymbol{u}}\|_{\ell_2}\left(\boldsymbol{u}^T\boldsymbol{A}_{\mathscr{S}_\tau^c}^T\boldsymbol{A}_{\mathscr{S}_\tau^c}\boldsymbol{w}\right)\\ &= \mu_\tau \|\boldsymbol{X}\tilde{\boldsymbol{u}}\|_{\ell_2}(\boldsymbol{u}_\perp^T\boldsymbol{A}_{\mathscr{S}_\tau^c}^T\boldsymbol{A}_{\mathscr{S}_\tau^c}\boldsymbol{w}) + \mu_\tau\frac{\tilde{\boldsymbol{u}}^T\boldsymbol{X}^T\boldsymbol{w}}{\|\boldsymbol{w}\|_{\ell_2}^2}\left\|\boldsymbol{A}_{\mathscr{S}_\tau^c}\boldsymbol{w}\right\|_{\ell_2}^2.\end{aligned}$$

Thus, using $\mu_\tau = \tilde{\mu}_\tau/\beta_{s,m}^2$ we have

$$\begin{aligned}&\mu_\tau \cdot \sup_{\tilde{\boldsymbol{u}}\in-\tilde{\mathscr{C}}\cap\mathbb{S}^{d-1}} \tilde{\boldsymbol{u}}^T\boldsymbol{X}^T\boldsymbol{A}_{\mathscr{S}_\tau^c}^T\boldsymbol{A}_{\mathscr{S}_\tau^c}\boldsymbol{w}\\ &\leqslant \frac{\tilde{\mu}_\tau}{\beta_{s_\tau,m}^2}\cdot\sup_{\tilde{\boldsymbol{u}}\in-\tilde{\mathscr{C}}\cap\mathbb{S}^{d-1}}\|\boldsymbol{X}\tilde{\boldsymbol{u}}\|_{\ell_2}\cdot\sup_{\boldsymbol{u}\in-\mathscr{C}\cap\mathbb{S}^{n-1}}\boldsymbol{u}_\perp^T\boldsymbol{A}_{\mathscr{S}_\tau^c}^T\boldsymbol{A}_{\mathscr{S}_\tau^c}\boldsymbol{w}\\ &\quad+\frac{\left\|\boldsymbol{A}_{\mathscr{S}_\tau^c}\boldsymbol{w}\right\|_{\ell_2}^2}{\beta_{s_\tau,m}^2\cdot\|\boldsymbol{w}\|_{\ell_2}^2}\cdot\tilde{\mu}_\tau\cdot\left(\sup_{\tilde{\boldsymbol{u}}\in\tilde{\mathscr{C}}\cap\mathbb{S}^{d-1}}\tilde{\boldsymbol{u}}^T\boldsymbol{X}^T\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_{\ell_2}}\right)\|\boldsymbol{w}\|_{\ell_2}\\ &= \frac{\tilde{\mu}_\tau\cdot\sigma_{\mathscr{R}}(\boldsymbol{X})}{\beta_{s_\tau,m}^2}\cdot\sup_{\boldsymbol{u}\in-\mathscr{C}\cap\mathbb{S}^{n-1}}\boldsymbol{u}_\perp^T\boldsymbol{A}_{\mathscr{S}_\tau^c}^T\boldsymbol{A}_{\mathscr{S}_\tau^c}\boldsymbol{w}+\frac{1}{\beta_{s_\tau,m}^2}\frac{\left\|\boldsymbol{A}_{\mathscr{S}_\tau^c}\boldsymbol{w}\right\|_{\ell_2}^2}{\|\boldsymbol{w}\|_{\ell_2}^2}\cdot\tilde{\mu}_\tau\cdot\xi(\boldsymbol{X})\|\boldsymbol{w}\|_{\ell_2}.\end{aligned}\tag{5.3}$$

All that remains is to bound the extra additive term in (5.2) and the extra additive and multiplicative terms in (5.3). To this aim note that for any γ_τ, we have

$$\boldsymbol{u}^T\left(\boldsymbol{I}-\gamma_\tau\boldsymbol{A}_{\mathscr{S}_\tau^c}^T\boldsymbol{A}_{\mathscr{S}_\tau^c}\right)\boldsymbol{v} = \frac{1}{4}\left(\|\boldsymbol{u}+\boldsymbol{v}\|_{\ell_2}^2-\gamma_\tau\left\|\boldsymbol{A}_{\mathscr{S}_\tau^c}(\boldsymbol{u}+\boldsymbol{v})\right\|_{\ell_2}^2\right)+\frac{1}{4}\left(\gamma_\tau\left\|\boldsymbol{A}_{\mathscr{S}_\tau^c}(\boldsymbol{u}-\boldsymbol{v})\right\|_{\ell_2}^2-\|\boldsymbol{u}-\boldsymbol{v}\|_{\ell_2}^2\right).\tag{5.4}$$

To proceed, we state a lemma about bounding set-restricted eigenvalues, which is proved in the Appendix section.

Step 2: In this step, we bound the deviation terms in (5.4). To this aim, we state a lemma regarding concentration of quadratic stochastic processes adjusted so as to deal with the effect of worst-case stragglers.

LEMMA 5.1 Let $\mathscr{T} \in \mathbb{R}^n$ and define $\sigma(\mathscr{T}) := \sup_{\boldsymbol{v}\in\mathscr{T}}\|\boldsymbol{v}\|_{\ell_2}$. Also assume the random encoding matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is a matrix with i.i.d. $\mathscr{N}(0,1)$ entries. Furthermore, define $\alpha_{s,m} = \sqrt{m-2-5s\log\left(\frac{em}{s}\right)}$ and

$\beta_{s,m} = \min\left(\sqrt{3(m-s)\log\left(\frac{em}{(m-s)}\right)}, \sqrt{m}\right)$. Then for all $\boldsymbol{u} \in \mathscr{T}$

$$\sup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s} \left\|\boldsymbol{A}_{\mathscr{S}^c}\boldsymbol{u}\right\|_{\ell_2} \leqslant \beta_{s,m}\|\boldsymbol{u}\|_{\ell_2} + (\omega(\mathscr{T}) + \eta), \tag{5.5}$$

holds with probability at least $1 - 2e^{-\frac{\eta^2}{8\sigma^2(\mathscr{T})}}$. Furthermore, for all $\boldsymbol{u} \in \mathscr{T}$

$$\inf_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s} \left\|\boldsymbol{A}_{\mathscr{S}^c}\boldsymbol{u}\right\|_{\ell_2} \geqslant \alpha_{s,m}\|\boldsymbol{u}\|_{\ell_2} - (\omega(\mathscr{T}) + \eta), \tag{5.6}$$

holds with probability at least $1 - 4e^{-\frac{\eta^2}{8\sigma^2(\mathscr{T})}}$.

This lemma relates the concentration of the Gaussian process $\boldsymbol{A}_{\mathscr{S}^c}\boldsymbol{u}$ to the Gaussian width of the set which $\boldsymbol{u}$ belongs to. By having a small corresponding Gaussian width, the concentration bound would be tighter.

Step 3: In this step, we utilize Lemma 5.1 to complete the bound 5.4 and provide convergence guarantees for the coded iterations (2.4). To use the above lemma, define $\mathscr{T}_+ = (\mathscr{C}\cap\mathbb{S}^{n-1}) + (\mathscr{C}\cap\mathbb{S}^{n-1})$ and $\mathscr{T}_- = (\mathscr{C}\cap\mathbb{S}^{n-1}) - (\mathscr{C}\cap\mathbb{S}^{n-1})$. Also note that $\sigma(\mathscr{T}_-) \leqslant 2$, $\sigma(\mathscr{T}_+) \leqslant 2$, $\omega(\mathscr{T}_-) \leqslant 2\omega(\mathscr{C}\cap\mathbb{S}^{n-1})$, $\omega(\mathscr{T}_+) \leqslant 2\omega(\mathscr{C}\cap\mathbb{S}^{n-1})$ and $\boldsymbol{u}+\boldsymbol{v} \in \mathscr{T}_+$. Thus, by Lemma 5.1 equation (5.6)

$$\begin{aligned}
\|\boldsymbol{u}+\boldsymbol{v}\|_{\ell_2}^2 - \gamma_\tau\left\|\boldsymbol{A}_{\mathscr{S}_\tau^c}(\boldsymbol{u}+\boldsymbol{v})\right\|_{\ell_2}^2 &\leqslant \|\boldsymbol{u}+\boldsymbol{v}\|_{\ell_2}^2 - \gamma_\tau\left(\alpha_{s_\tau,m}\|\boldsymbol{u}+\boldsymbol{v}\|_{\ell_2} - \left(\omega(\mathscr{T}_+)+\eta\right)\right)^2 \\
&\leqslant \left(1-\gamma_\tau\alpha_{s_\tau,m}^2\right)\|\boldsymbol{u}+\boldsymbol{v}\|_{\ell_2}^2 \\
&\quad + 2\gamma_\tau\alpha_{s_\tau,m}\left(\omega(\mathscr{T}_+)+\eta\right)\|\boldsymbol{u}+\boldsymbol{v}\|_{\ell_2} - \gamma_\tau\left(\omega(\mathscr{T}_+)+\eta\right)^2 \\
&= \left(1-\gamma_\tau\alpha_{s_\tau,m}^2\right)\|\boldsymbol{u}+\boldsymbol{v}\|_{\ell_2}^2 \\
&\quad + 2\gamma_\tau\alpha_{s_\tau,m}\left(2\omega(\mathscr{C}\cap\mathbb{S}^{n-1})+\eta\right)\|\boldsymbol{u}+\boldsymbol{v}\|_{\ell_2} \\
&\quad - \gamma_\tau\left(2\omega(\mathscr{C}\cap\mathbb{S}^{n-1})+\eta\right)^2
\end{aligned} \tag{5.7}$$

holds with probability at least $1-4e^{-\frac{\eta^2}{32}}$. Also, $\boldsymbol{u}-\boldsymbol{v}\in\mathscr{T}_-$, thus by Lemma 5.1 equation (5.5)

$$\begin{aligned}\gamma_\tau\left\|\boldsymbol{A}_{\mathscr{S}_\tau^c}(\boldsymbol{u}-\boldsymbol{v})\right\|_{\ell_2}^2-\|\boldsymbol{u}-\boldsymbol{v}\|_{\ell_2}^2 &\leqslant \left(\gamma_\tau\beta_{s_\tau,m}^2-1\right)\|\boldsymbol{u}-\boldsymbol{v}\|_{\ell_2}^2+2\gamma_\tau\beta_{s_\tau,m}\left(\omega(\mathscr{T}_-)+\eta\right)\|\boldsymbol{u}-\boldsymbol{v}\|_{\ell_2}\\ &\quad+\gamma_\tau\left(\omega(\mathscr{T}_-)+\eta\right)^2\\ &=\left(\gamma_\tau\beta_{s_\tau,m}^2-1\right)\|\boldsymbol{u}-\boldsymbol{v}\|_{\ell_2}^2+2\gamma_\tau\beta_{s_\tau,m}\left(2\omega(\mathscr{C}\cap\mathbb{S}^{n-1})+\eta\right)\|\boldsymbol{u}-\boldsymbol{v}\|_{\ell_2}\\ &\quad+\gamma_\tau\left(2\omega(\mathscr{C}\cap\mathbb{S}^{n-1})+\eta\right)^2 \qquad (5.8)\end{aligned}$$

holds with probability at least $1-2e^{-\frac{\eta^2}{32}}$. Plugging these bounds into (5.4) with $\|\boldsymbol{u}+\boldsymbol{v}\|_{\ell_2}\leqslant 2$ and $\|\boldsymbol{u}-\boldsymbol{v}\|_{\ell_2}\leqslant 2$ and using the short-hand $\omega:=\omega(\mathscr{C}\cap\mathbb{S}^{n-1})$, we conclude that for $\gamma_\tau=\frac{1}{\beta_{s_\tau,m}^2}$

$$\begin{aligned}\boldsymbol{u}^T(\boldsymbol{I}-\gamma_\tau\boldsymbol{A}_{\mathscr{S}_\tau^c}^T\boldsymbol{A}_{\mathscr{S}_\tau^c})\boldsymbol{v} &\leqslant \frac{1}{4}\left(1-\gamma_\tau\alpha_{s_\tau,m}^2\right)\|\boldsymbol{u}+\boldsymbol{v}\|_{\ell_2}^2+\frac{1}{2}\gamma_\tau\alpha_{s_\tau,m}\left(2\omega+\eta\right)\|\boldsymbol{u}+\boldsymbol{v}\|_{\ell_2}\\ &\quad+\frac{1}{4}\left(\gamma_\tau\beta_{s_\tau,m}^2-1\right)\|\boldsymbol{u}-\boldsymbol{v}\|_{\ell_2}^2+\frac{1}{2}\gamma_\tau\beta_{s_\tau,m}\left(2\omega+\eta\right)\|\boldsymbol{u}-\boldsymbol{v}\|_{\ell_2}\\ &\leqslant 1-\frac{\alpha_{s,m}^2}{\beta_{s_\tau,m}^2}+4\frac{\left(\omega+\frac{\eta}{2}\right)}{\beta_{s_\tau,m}} \qquad (5.9)\end{aligned}$$

holds with probability at least $1-6e^{-\frac{\eta^2}{32}}$. Using a change of variable η to 2η together with the fact that $\sqrt{m-s_\tau}\leqslant\beta_{s_\tau,m}\leqslant\sqrt{m}$, we arrive at

$$\boldsymbol{u}^T(\boldsymbol{I}-\mu_\tau\boldsymbol{A}_{\mathscr{S}_\tau^c}^T\boldsymbol{A}_{\mathscr{S}_\tau^c})\boldsymbol{v}\leq\frac{2+5s_\tau\log(em/s_\tau)}{m}+4\sqrt{\frac{m_0}{m-s_\tau}},$$

holds with probability at least $1-6e^{-\frac{\eta^2}{8}}$ completing the proof of the bound on the extra term of (5.2).

Now we focus on the extra additive and multiplicative terms in (5.3). We begin with the additive term. To this aim note that since $\boldsymbol{u}_\perp$ is orthogonal to $\boldsymbol{w}$, $\boldsymbol{u}_\perp^T\boldsymbol{A}_{\mathscr{S}_\tau^c}^T\boldsymbol{A}_{\mathscr{S}_\tau^c}\boldsymbol{w}$ has the same distribution as $\|\boldsymbol{w}\|_{\ell_2}\boldsymbol{u}_\perp^T\boldsymbol{A}_{\mathscr{S}_\tau^c}^T\boldsymbol{a}$ with $\boldsymbol{a}$ distributed as $\boldsymbol{a}\sim\mathcal{N}(0,\boldsymbol{I}_{m-|\mathscr{S}_\tau|})$ and independent from $\boldsymbol{A}$. Similarly, $\|\boldsymbol{w}\|_{\ell_2}\boldsymbol{u}_\perp^T\boldsymbol{A}_{\mathscr{S}_\tau^c}^T\boldsymbol{a}$ has the same distribution as $\|\boldsymbol{a}\|_{\ell_2}\|\boldsymbol{w}\|_{\ell_2}(\boldsymbol{u}_\perp^T\boldsymbol{g})$ with $\boldsymbol{g}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_n)$. Therefore,

$$\begin{aligned}\sup_{\boldsymbol{u}\in-\mathscr{C}\cap\mathbb{S}^{n-1}}\boldsymbol{u}_\perp^T\boldsymbol{A}_{\mathscr{S}_\tau^c}^T\boldsymbol{A}_{\mathscr{S}_\tau^c}\boldsymbol{w} &= \|\boldsymbol{a}\|_{\ell_2}\|\boldsymbol{w}\|_{\ell_2}\cdot\left(\sup_{\boldsymbol{u}\in-\mathscr{C}\cap\mathbb{S}^{n-1}}\boldsymbol{u}_\perp^T\boldsymbol{g}\right)\\ &\leqslant \|\boldsymbol{a}\|_{\ell_2}\|\boldsymbol{w}\|_{\ell_2}\cdot(\omega+\eta)\\ &\leqslant \sqrt{2(m-|\mathscr{S}|_\tau)}\,\|\boldsymbol{w}\|_{\ell_2}\cdot(\omega+\eta) \qquad (5.10)\end{aligned}$$

holds with probability at least $1-e^{-\frac{\eta^2}{2}}-e^{-\frac{m}{2}}$.

We now focus on the extra multiplicative term in (5.3). To this aim note that since $\boldsymbol{w}$ is fixed $\boldsymbol{A}_{\mathscr{S}_\tau^c}\boldsymbol{w}$ is distributed as $\|\boldsymbol{w}\|_{\ell_2}\boldsymbol{a}$ with $\boldsymbol{a}$ distributed as $\mathcal{N}(0, \boldsymbol{I}_{m-|\mathscr{S}_\tau|})$. Therefore,

$$\frac{\left\|\boldsymbol{A}_{\mathscr{S}_\tau^c}\boldsymbol{w}\right\|_{\ell_2}^2}{\|\boldsymbol{w}\|_{\ell_2}^2} = \|\boldsymbol{a}\|_{\ell_2}^2 \leqslant 2(m - |\mathscr{S}_\tau|), \tag{5.11}$$

holds with probability at least $1 - e^{-\frac{m}{2}}$. Plugging (5.10) and (5.11) into (5.3), we conclude that

$$\begin{aligned}
\mu_\tau \cdot \sup_{\tilde{\boldsymbol{u}} \in -\widetilde{\mathscr{C}} \cap \mathbb{S}^{d-1}} \tilde{\boldsymbol{u}}^T \boldsymbol{X}^T \boldsymbol{A}_{\mathscr{S}_\tau^c}^T \boldsymbol{A}_{\mathscr{S}_\tau^c} \boldsymbol{w} &\leqslant \sqrt{2}\frac{\tilde{\mu}_\tau \cdot \sigma_{\mathscr{R}}(\boldsymbol{X})}{\beta_{s_\tau,m}^2} \cdot \sqrt{m - |\mathscr{S}_\tau|}\, \|\boldsymbol{w}\|_{\ell_2} \cdot (\omega + \eta) \\
&\quad + \frac{2(m - |\mathscr{S}_\tau|)}{\beta_{s_\tau,m}^2} \cdot \tilde{\mu}_\tau \cdot \xi(\boldsymbol{X})\, \|\boldsymbol{w}\|_{\ell_2} \\
&\leqslant \tilde{\mu}_\tau \cdot \sigma_{\mathscr{R}}(\boldsymbol{X}) \sqrt{\frac{m_0}{2(m - s_\tau)\log^2\left(\frac{em}{m - s_\tau}\right)}}\, \|\boldsymbol{w}\|_{\ell_2} \\
&\quad + \frac{1}{\log\left(\frac{em}{m - s_\tau}\right)} \tilde{\mu}_\tau \cdot \xi(\boldsymbol{X})\, \|\boldsymbol{w}\|_{\ell_2} \\
&\leqslant \tilde{\mu}_\tau \cdot \xi(\boldsymbol{X})\, \|\boldsymbol{w}\|_{\ell_2} + \frac{1}{\sqrt{2}} \tilde{\mu} \cdot \sigma_{\mathscr{R}}(\boldsymbol{X}) \sqrt{\frac{m_0}{m - s_\tau}},
\end{aligned} \tag{5.12}$$

holds with probability at least $1 - e^{-\frac{\eta^2}{2}} - e^{-\frac{m}{2}}$.

Funding

M. Soltanolkotabi is supported by the Packard Fellowship in Science and Engineering, a Sloan Research Fellowship in Mathematics, National Science Foundation Computing and Communication Foundations-Communications and Information Foundations grants 1846369 and 1813877, Air Force Office of Scientific Research-Young Investigator Program under award FA9550-18-1-0078, Defense Advanced Research Projects Agency Learning with Less Labels (LwLL) and Fast Network Interface Cards (FastNICs) programs and a Google faculty research award (to M.S.). Part of the work was done while visiting the Simons Institute for the Theory of Computing.

Data availability statement

This paper contains synthetic generated data which has been reported.

References

1. Ananthanarayanan, G., Kandula, S., Greenberg, A., Stoica, I., Lu, Y., Saha, B. & Harris, E. (2010) Reining in the Outliers in Map-reduce Clusters Using Mantri. *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation*. Berkeley, CA United States: USENIX Association.

2. Charles, Z. B., Papailiopoulos, D. S. & Ellenberg, J. S. (2017) Approximate gradient coding via sparse random graphs. ArXiv, abs/1711.06771.
3. Dean, J. & Barroso, L. A. (2013) The tail at scale. *Commun. ACM*, **56**, 74–80.
4. Dutta, S., Cadambe, V. & Grover, P. (2016) Short-dot: computing large linear transforms distributedly using coded short dot products. *Advances in Neural Information Processing Systems*. Barcelona, Spain: NIPS 2016, Centre Convencions Internacional Barcelona pp. 2092–2100.
5. Gordon, Y. (1988) *On Milman's Inequality and Random Subspaces which Escape Through a Mesh in R^n*. Springer. https://link.springer.com/chapter/10.1007/BFb0081737, Springer book.
6. Halbawi, W., Ruhi, N. A., Salehi, F. & Hassibi, B. (2017) Improving distributed gradient descent using Reed–Solomon codes. *arXiv:1706.05436*.
7. Karakus, C., Sun, Y. & Diggavi, S. (2017) *Encoded distributed optimization*. 2017 IEEE International Symposium on Information Theory (ISIT). Aachen, Germany: ISIT 2017, pp. 2890–2894.
8. Karakus, C., Sun, Y., Diggavi, S. & Yin, W. (2017) Straggler mitigation in distributed optimization through data encoding. *Advances in Neural Information Processing Systems 30*. pp. 5440–5448.
9. Ledoux, M. & Talagrand, M. (1991) *Probability in Banach Spaces: Isoperimetry and Processes*. Springer. https://link.springer.com/book/10.1007/978-3-642-20212-4, Springer book.
10. Lee, K., Lam, M., Pedarsani, R., Papailiopoulos, D. & Ramchandran, K. (2016) Speeding up distributed machine learning using codes. *2016 IEEE International Symposium on Information Theory (ISIT)*. Barcelona: ISIT 2016, pp. 1143–1147.
11. Li, S., Kalan, S. M. M., Yu, Q., Soltanolkotabi, M. & Avestimehr, A. S. (2018) Polynomially coded regression: optimal straggler mitigation via data encoding. *arXiv preprint, arXiv:1805.09934*.
12. Li, S., Maddah-Ali, M. A. & Avestimehr, A. S. (2016) A unified coding framework for distributed computing with straggling servers. 2016 IEEE Globecom Workshops (GC Wkshps). 1–6.
13. Li, S., Maddah-Ali, M. A., Yu, Q. & Avestimehr, A. S. (2018) A fundamental tradeoff between computation and communication in distributed computing. *IEEE Trans. Inf. Theory*, **64**, 109–128.
14. Oymak, S., Recht, B. & Soltanolkotabi, M. (2017) *Sharp time-data tradeoffs for linear inverse problems*. IEEE Transactions on Information Theory, **64**, 4129–4158.
15. Oymak, S. & Soltanolkotabi, M. (2016) Fast and reliable parameter estimation from nonlinear observations. *SIAM J. Optim.* **27**, 2276–2300.
16. Pilanci, M. & Wainwright, M. J. (2015) Randomized sketches of convex programs with sharp guarantees. *IEEE Trans. Inf. Theory*, **61**, 5096–5115.
17. Qi, H., Sparks, E. R. & Talwalkar, A. (2017) Paleo: a performance model for deep neural networks. Toulon, France: ICLR 2017, Palais des Congrés Neptune.
18. Raviv, N., Tamo, I., Tandon, R. & Dimakis, A. G. (2018) Gradient coding from cyclic MDS codes and expander graphs. International Conference on Machine Learning. 4305–4313.
19. Recht, B., Re, C., Wright, S. & Niu, F. (2011) Hogwild: a lock-free approach to parallelizing stochastic gradient descent. *Advances in Neural Information Processing Systems*. Granada, Spain: NIPS 2011, pp. 693–701.
20. Tanond, R., Lei, Q., Dimakis, A. G. & Karampatziakis, N. (2017) Gradient coding: avoiding stragglers in distributed learning. *Proceedings of the 34th ICML*. PMLR. Sydney, Australia: ICML 2017, International Convention Centre
21. Wang, H., Charles, Z. B. & Papailiopoulos, D. S. (2019) ErasureHead: Distributed gradient descent without delays using approximate gradient coding. CoRR, arXiv preprint arXiv:1901.09671.
22. Yu, Q., Maddah-Ali, M. & Avestimehr, S. (2017) Polynomial codes: an optimal design for high-dimensional coded matrix multiplication. *Advances in Neural Information Processing Systems 30*. Long Beach: NIPS 2017, Long Beach Convention Center, Curran Associates, Inc., pp. 4406–4416.
23. Zaharia, M., Konwinski, A., Joseph, A. D., Katz, R. H. & Stoica, I. (2008) Improving MapReduce performance in heterogeneous environments. San Diego: OSDI 2008, vol. 8, p. 7.

24. Zhang, X., Liu, J., Zhu, Z. & Bentley, E. S. (2020) Communication-efficient network-distributed optimization with differential-coded compressors. IEEE INFOCOM 2020 - IEEE Conference on Computer Communications. 317–326.

25. Zhu, J., Pu, Y., Gupta, V., Tomlin, C. & Ramchandran, K. (2017) A sequential approximation framework for coded distributed optimization. *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IL, USA: 2017 Allerton Conference, Allerton Park and Retreat Center, Monticello, pp. 1240–1247.

Appendix A

In this section, we aim to prove Lemma 5.1 stated in the proofs section. Our proof is related to the proof of Gordon's celebrated escape through the mesh [**5**, Theorem A]. We will first show the bound (5.5). To this aim, we make use of Slepian's lemma stated below.

Lemma A.1 (Slepian's inequality). [**9**, Section 3.3] If X_t and Y_t are a.s. bounded, Gaussian processes on T such that $\mathbb{E}[X_t] = \mathbb{E}[Y_t]$ and $\mathbb{E}[X_t^2] = \mathbb{E}[Y_t^2]$ for all $t \in T$ and

$$\mathbb{E}\left[(X_t - X_s)^2\right] \leq \mathbb{E}\left[(Y_t - Y_s)^2\right],$$

for all $s, t \in T$, then for all real t,

$$\mathbb{E}\left[\sup_{t\in T} X_t\right] \leq \mathbb{E}\left[\sup_{t\in T} Y_t\right]. \tag{A.1}$$

Furthermore,

$$\mathbb{P}\left\{\bigcup_{t\in T}[X_t > \eta_t]\right\} \leq \mathbb{P}\left\{\bigcup_{t\in T}[Y_t > \eta_t]\right\}. \tag{A.2}$$

Define $\boldsymbol{I}_{\mathscr{S}^c} \in \mathbb{R}^{(m-s)\times n}$ as the part of the identity matrix that keeps the rows indexed by $\mathscr{S}^c$. For $\boldsymbol{u} \in \mathscr{T}$ and $\boldsymbol{v} \in \mathbb{S}^{m-s-1} = \{\boldsymbol{v} \in \mathbb{R}^{m-s};\ \|\boldsymbol{v}\|_{\ell_2} = 1\}$, we define three Gaussian processes:

$$X_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} = \boldsymbol{v}^*\boldsymbol{I}_{\mathscr{S}^c}\boldsymbol{A}\boldsymbol{u}, \quad Y_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} = \|\boldsymbol{u}\|_{\ell_2}\,\boldsymbol{v}^*\boldsymbol{I}_{\mathscr{S}^c}\boldsymbol{a} + \boldsymbol{g}^*\boldsymbol{u} \quad \text{and} \quad Z_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} = \|\boldsymbol{u}\|_{\ell_2}\,(\boldsymbol{v}^*\boldsymbol{I}_{\mathscr{S}^c}\boldsymbol{a} - \beta_{s,m}) + \boldsymbol{g}^*\boldsymbol{u}.$$

Here $\boldsymbol{a} \in \mathbb{R}^m$ is distributed as $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_m)$ and $\boldsymbol{g} \in \mathbb{R}^n$ is distributed as $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$. It follows that for all $\boldsymbol{u}, \tilde{\boldsymbol{u}} \in \mathcal{T}, \boldsymbol{v}, \tilde{\boldsymbol{v}} \in \mathbb{S}^{m-s-1}$ and $\mathcal{S}, \tilde{\mathcal{S}} \subset \{1, 2, \ldots, m\}$, we have

$$
\begin{aligned}
&\mathbb{E}\left|Y_{(\boldsymbol{u},\mathcal{S}),\boldsymbol{v}} - Y_{(\tilde{\boldsymbol{u}},\tilde{\mathcal{S}}),\tilde{\boldsymbol{v}}}\right|^2 - \mathbb{E}\left|X_{(\boldsymbol{u},\mathcal{S}),\boldsymbol{v}} - X_{(\tilde{\boldsymbol{u}},\tilde{\mathcal{S}}),\tilde{\boldsymbol{v}}}\right|^2 \\
&= \left\| \|\boldsymbol{u}\|_{\ell_2} \boldsymbol{I}^T_{\mathcal{S}^c}\boldsymbol{v} - \|\tilde{\boldsymbol{u}}\|_{\ell_2} \boldsymbol{I}^T_{\tilde{\mathcal{S}}^c}\tilde{\boldsymbol{v}} \right\|^2_{\ell_2} + \|\boldsymbol{u} - \tilde{\boldsymbol{u}}\|^2_{\ell_2} - \left\| \boldsymbol{u}\left(\boldsymbol{I}^T_{\mathcal{S}^c}\boldsymbol{v}\right)^T - \tilde{\boldsymbol{u}}\left(\boldsymbol{I}^T_{\tilde{\mathcal{S}}^c}\tilde{\boldsymbol{v}}\right)^T \right\|^2_F \\
&= \left(\|\boldsymbol{u}\|^2_{\ell_2} + \|\tilde{\boldsymbol{u}}\|^2_{\ell_2}\right) - 2\|\boldsymbol{u}\|_{\ell_2}\|\tilde{\boldsymbol{u}}\|_{\ell_2}\langle \boldsymbol{I}^T_{\mathcal{S}^c}\boldsymbol{v}, \boldsymbol{I}^T_{\tilde{\mathcal{S}}^c}\tilde{\boldsymbol{v}}\rangle - 2\langle \boldsymbol{u}, \tilde{\boldsymbol{u}}\rangle\langle \boldsymbol{I}^T_{\mathcal{S}^c}\boldsymbol{v}, \boldsymbol{I}^T_{\tilde{\mathcal{S}}^c}\tilde{\boldsymbol{v}}\rangle \\
&= \left(\|\boldsymbol{u}\|^2_{\ell_2} + \|\tilde{\boldsymbol{u}}\|^2_{\ell_2}\right) - 2\|\boldsymbol{u}\|_{\ell_2}\|\tilde{\boldsymbol{u}}\|_{\ell_2}\langle \boldsymbol{I}^T_{\mathcal{S}^c}\boldsymbol{v}, \boldsymbol{I}^T_{\tilde{\mathcal{S}}^c}\tilde{\boldsymbol{v}}\rangle - 2\langle \boldsymbol{u}, \tilde{\boldsymbol{u}}\rangle\left(1 - \langle \boldsymbol{I}^T_{\mathcal{S}^c}\boldsymbol{v}, \boldsymbol{I}^T_{\tilde{\mathcal{S}}^c}\tilde{\boldsymbol{v}}\rangle\right) \\
&\geq 2\|\boldsymbol{u}\|_{\ell_2}\|\tilde{\boldsymbol{u}}\|_{\ell_2} - 2\|\boldsymbol{u}\|_{\ell_2}\|\tilde{\boldsymbol{u}}\|_{\ell_2}\langle \boldsymbol{I}^T_{\mathcal{S}^c}\boldsymbol{v}, \boldsymbol{I}^T_{\tilde{\mathcal{S}}^c}\tilde{\boldsymbol{v}}\rangle - 2\langle \boldsymbol{u}, \tilde{\boldsymbol{u}}\rangle\left(1 - \langle \boldsymbol{I}^T_{\mathcal{S}^c}\boldsymbol{v}, \boldsymbol{I}^T_{\tilde{\mathcal{S}}^c}\tilde{\boldsymbol{v}}\rangle\right) \\
&= 2\left(\|\boldsymbol{u}\|_{\ell_2}\|\tilde{\boldsymbol{u}}\|_{\ell_2} - \langle \boldsymbol{u}, \tilde{\boldsymbol{u}}\rangle\right)\left(1 - \langle \boldsymbol{I}^T_{\mathcal{S}^c}\boldsymbol{v}, \boldsymbol{I}^T_{\tilde{\mathcal{S}}^c}\tilde{\boldsymbol{v}}\rangle\right) \\
&\geq 0. \qquad \text{(A.3)}
\end{aligned}
$$

AIt is trivial to check that $\mathbb{E}[X_{(\boldsymbol{u},\mathcal{S}),\boldsymbol{v}}] = \mathbb{E}[Y_{(\boldsymbol{u},\mathcal{S}),\boldsymbol{v}}]$ and $\mathbb{E}[X^2_{(\boldsymbol{u},\mathcal{S}),\boldsymbol{v}}] = \mathbb{E}[Y^2_{(\boldsymbol{u},\mathcal{S}),\boldsymbol{v}}]$ for all $\boldsymbol{u} \in \mathcal{T}, \boldsymbol{v} \in \mathbb{S}^{m-s-1}$ and $\mathcal{S} \subset \{1, 2, \ldots, m\}$. Thus, the two Gaussian processes $X_{(\boldsymbol{u},\mathcal{S}),\boldsymbol{v}}$ and $Y_{(\boldsymbol{u},\mathcal{S}),\boldsymbol{v}}$ obey the three assumptions of Slepian's inequality.

Now define the function $f(\boldsymbol{x}) = \sup_{\mathcal{S}\subset\{1,2,\ldots,m\},\ |\mathcal{S}|=s} \|\boldsymbol{x}_{\mathcal{S}^c}\|_{\ell_2}$ and let

$$
\mathcal{S}^c_{\boldsymbol{x}} = \underset{\mathcal{S}\subset\{1,2,\ldots,m\},\ |\mathcal{S}|=s}{\arg\max} f(\boldsymbol{x}) \quad \text{and} \quad \mathcal{S}^c_{\boldsymbol{y}} = \underset{\mathcal{S}\subset\{1,2,\ldots,m\},\ |\mathcal{S}|=s}{\arg\max} f(\boldsymbol{y}).
$$

We wish to bound $f(\boldsymbol{a})$ with high probability. To this aim first note that by concentration of Lipschitz functions of Gaussians

$$
\mathbb{P}\left\{ \|\boldsymbol{a}_{\mathcal{S}^c}\|_{\ell_2} - \mathbb{E}[\|\boldsymbol{a}_{\mathcal{S}^c}\|_{\ell_2} \geq \delta \right\} \leqslant e^{\frac{-\delta^2}{2}}.
$$

Note that since $\mathbb{E}[\|\boldsymbol{a}_{\mathcal{S}^c}\|_{\ell_2}] \leqslant \sqrt{\mathbb{E}[\|\boldsymbol{a}_{\mathcal{S}^c}\|^2_{\ell_2}]} = \sqrt{m-s}$, by substituting $\delta = \eta + \sqrt{2(m-s)\log\left(\frac{em}{m-s}\right)}$, we have

$$
\begin{aligned}
\mathbb{P}\left\{ \|\boldsymbol{a}_{\mathcal{S}^c}\|_{\ell_2} - \sqrt{m-s} \geq \eta + \sqrt{2(m-s)\log\left(\frac{em}{m-s}\right)} \right\} &\leqslant e^{\frac{-\left(\eta+\sqrt{2(m-s)\log\left(\frac{em}{m-s}\right)}\right)^2}{2}} \\
&\leqslant e^{\frac{-\left(\sqrt{2(m-s)\log\left(\frac{em}{m-ms}\right)}\right)^2}{2}} e^{\frac{-\eta^2}{2}}.
\end{aligned}
$$

Using union bound, we have

$$\begin{aligned}
&\mathbb{P}\left\{\sup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s} \left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2} \geq \eta+\sqrt{m-s}+\sqrt{2(m-s)\log\left(\frac{em}{m-s}\right)}\right\}\\
&\leqslant \binom{m}{m-s} e^{\frac{-\left(\sqrt{2(m-s)\log\left(\frac{em}{m-s}\right)}\right)^2}{2}} e^{\frac{-\eta^2}{2}}\\
&= \binom{m}{m-s}\left(\frac{em}{m-s}\right)^{-s} e^{\frac{-\eta^2}{2}}\\
&\leqslant e^{\frac{-\eta^2}{2}}.
\end{aligned}$$

We thus conclude that

$$\mathbb{P}\left\{\sup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s} \left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2} \geq \eta+\sqrt{m-s}+\sqrt{2(m-s)\log\left(\frac{em}{m-s}\right)}\right\} \leqslant e^{\frac{-\eta^2}{2}}. \tag{A.4}$$

Also note that

$$\sqrt{m-s}+\sqrt{2(m-s)\log\left(\frac{em}{m-s}\right)} \leqslant \min\left(\sqrt{3(m-s)\log\left(\frac{em}{m-s}\right)}, m\right) := \beta_{s,m}.$$

The latter together with (A4) allows us to conclude that

$$\mathbb{P}\left\{f(\boldsymbol{a}) \geqslant \beta_{s,m}+\eta\right\} \leqslant e^{-\frac{\eta^2}{2}}. \tag{A.5}$$

Now consider the relationship of following sets:

$$\begin{aligned}
\left\{\boldsymbol{a}:\quad \|\boldsymbol{u}\|_{\ell_2} f(\boldsymbol{a}) \geqslant \|\boldsymbol{u}\|_{\ell_2}\beta_{s,m}+\eta\right\} &\subset \left\{\boldsymbol{a}:\quad \|\boldsymbol{u}\|_{\ell_2} f(\boldsymbol{a}) \geqslant \|\boldsymbol{u}\|_{\ell_2}\beta_{s,m}+\|\boldsymbol{u}\|_{\ell_2}\frac{\eta}{\sigma(\mathscr{T})}\right\},\\
&= \left\{\boldsymbol{a}:\quad f(\boldsymbol{a}) \geqslant \beta_{s,m}+\frac{\eta}{\sigma(\mathscr{T})}\right\}.
\end{aligned}$$

Furthermore, note for every $\boldsymbol{u}\in\mathscr{T}$, $\left\{\boldsymbol{a}:\quad \|\boldsymbol{u}\|_{\ell_2} f(\boldsymbol{a}) \geqslant \|\boldsymbol{u}\|_{\ell_2}\beta_{s,m}+\eta\right\}$ is a subset of $\left\{\boldsymbol{a}:\quad f(\boldsymbol{a}) \geqslant \beta_{s,m}+\frac{\eta}{\sigma(\mathscr{T})}\right\}$. Combining the latter with (A.5), we arrive at

$$\mathbb{P}\left\{\bigcup_{\boldsymbol{u}\in\mathscr{T}}\left\{\boldsymbol{a}:\quad f(\boldsymbol{a})\,\|\boldsymbol{u}\|_{\ell_2} > \|\boldsymbol{u}\|_{\ell_2}\beta_{s,m}+\eta_1\right\}\right\} \leq \mathbb{P}\left\{\boldsymbol{a}:\ f(\boldsymbol{a}) \geqslant \beta_{s,m}+\frac{\eta_1}{\sigma(\mathscr{T})}\right\} \leqslant e^{-\frac{\eta_1^2}{2\sigma^2(\mathscr{T})}},$$

which immediately implies

$$\mathbb{P}\left\{\max_{\boldsymbol{u}\in\mathscr{T}}\ \|\boldsymbol{u}\|_{\ell_2}\left(f(\boldsymbol{a})-\beta_{s,m}\right) > \frac{\eta}{2}\right\} \leqslant e^{-\frac{\eta^2}{8\sigma^2(\mathscr{T})}}. \tag{A.6}$$

Also by the concentration of Lipschitz functions of Gaussians for the function $\max_{\boldsymbol{u}\in\mathscr{T}} (\boldsymbol{g}^*\boldsymbol{u})$ and the definition of Gaussian width, we have

$$\mathbb{P}\left\{ \max_{\boldsymbol{u}\in\mathscr{T}} \left(\boldsymbol{g}^*\boldsymbol{u}\right) > \omega(\mathscr{T}) + \frac{\eta}{2} \right\} = \mathbb{P}\left\{ \max_{\boldsymbol{u}\in\mathscr{T}} \left(\boldsymbol{g}^*\boldsymbol{u}\right) > \mathbb{E}\left[\max_{\boldsymbol{u}\in\mathscr{T}} \left(\boldsymbol{g}^*\boldsymbol{u}\right)\right] + \frac{\eta}{2} \right\} \leqslant e^{-\frac{\eta^2}{8\sigma^2(\mathscr{T})}}. \quad \text{(A.7)}$$

So far, we have obtained upper bounds on the probability of sets $\left\{\max_{\boldsymbol{u}\in\mathscr{T}} \|\boldsymbol{u}\|_{\ell_2} \left(f(\boldsymbol{a}) - \beta_{s,m}\right) > \frac{\eta}{2}\right\}$ and $\left\{ \max_{\boldsymbol{u}\in\mathscr{T}} (\boldsymbol{g}^*\boldsymbol{u}) > \omega(\mathscr{T}) + \frac{\eta}{2}\right\}$. In order to utilize these two sets, we combine them in the following way. Note that if

$$\max_{\boldsymbol{u}\in\mathscr{T}} \left(\|\boldsymbol{u}\|_{\ell_2} \left(f(\boldsymbol{a}) - \beta_{s,m}\right) + \boldsymbol{g}^*\boldsymbol{u}\right) > \eta + \omega(\mathscr{T}),$$

then we have either $\max_{\boldsymbol{u}\in\mathscr{T}} \|\boldsymbol{u}\|_{\ell_2} \left(f(\boldsymbol{a}) - \beta_{s,m}\right) > \frac{\eta}{2}$ or $\max_{\boldsymbol{u}\in\mathscr{T}} (\boldsymbol{g}^*\boldsymbol{u}) > \omega(\mathscr{T}) + \frac{\eta}{2}$, which implies that

$$\left\{\boldsymbol{a},\boldsymbol{g} : \max_{\boldsymbol{u}\in\mathscr{T}} \left(\|\boldsymbol{u}\|_{\ell_2} f(\boldsymbol{a}) + \boldsymbol{g}^*\boldsymbol{u} - \beta_{s,m}\|\boldsymbol{u}\|_{\ell_2}\right) > \omega(\mathscr{T}) + \eta\right\}$$

is a subset of

$$\left\{\boldsymbol{a},\boldsymbol{g} : \max_{\boldsymbol{u}\in\mathscr{T}} \|\boldsymbol{u}\|_{\ell_2} \left(f(\boldsymbol{a}) - \beta_{s,m}\right) > \frac{\eta}{2}\right\} \bigcup \left\{\boldsymbol{a},\boldsymbol{g} : \max_{\boldsymbol{u}\in\mathscr{T}} \left(\boldsymbol{g}^*\boldsymbol{u}\right) > \omega(\mathscr{T}) + \frac{\eta}{2}\right\}.$$

Using the latter together with (A.6) and (A.7) and using the independence of $\boldsymbol{a}$ and $\boldsymbol{g}$, we have

$$\begin{aligned}
&\mathbb{P}\left\{\max_{\boldsymbol{u}\in\mathscr{T}} \left(\|\boldsymbol{u}\|_{\ell_2} \left(f(\boldsymbol{a}) - \beta_{s,m}\right) + \boldsymbol{g}^*\boldsymbol{u}\right) > \omega(\mathscr{T}) + \eta\right\} \\
&\quad\leqslant \mathbb{P}\left\{\max_{\boldsymbol{u}\in\mathscr{T}} \|\boldsymbol{u}\|_{\ell_2} \left(f(\boldsymbol{a}) - \beta_{s,m}\right) > \frac{\eta}{2}\right\} + \mathbb{P}\left\{ \max_{\boldsymbol{u}\in\mathscr{T}} \left(\boldsymbol{g}^*\boldsymbol{u}\right) > \omega(\mathscr{T}) + \frac{\eta}{2}\right\} \\
&\quad\leqslant 2e^{-\frac{\eta^2}{8\sigma^2(\mathscr{T})}}.
\end{aligned}$$

Using the definition of $f(\boldsymbol{x}) = \sup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s} \|\boldsymbol{x}_{\mathscr{S}^c}\|_{\ell_2}$, the latter statement can be rewritten in the form

$$\mathbb{P}\left\{\sup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s} \max_{\boldsymbol{u}\in\mathscr{T}} \|\boldsymbol{u}\|_{\ell_2} \left(\left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2} - \beta_{s,m}\right) + \boldsymbol{g}^*\boldsymbol{u} > \omega(\mathscr{T}) + \eta\right\} \leqslant 2e^{-\frac{\eta^2}{8\sigma^2(\mathscr{T})}}. \quad \text{(A.8)}$$

As we defined the Gaussian process $Z_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} = \|\boldsymbol{u}\|_{\ell_2} (\boldsymbol{v}^*\boldsymbol{I}_{\mathscr{S}^c}\boldsymbol{a} - \beta_{s,m}) + \boldsymbol{g}^*\boldsymbol{u}$, we can write

$$\sup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s} \max_{\boldsymbol{u}\in\mathscr{T},\ \boldsymbol{v}\in\mathbb{S}^{m-s-1}} Z_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} = \sup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s} \max_{\boldsymbol{u}\in\mathscr{T}} \|\boldsymbol{u}\|_{\ell_2} \left(\left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2} - \beta_{s,m}\right) + \boldsymbol{g}^*\boldsymbol{u}.$$

This together with (A8) implies

$$\mathbb{P}\left\{\sup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s} \max_{\boldsymbol{u}\in\mathscr{T},\ \boldsymbol{v}\in\mathbb{S}^{m-s-1}} Z_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} > \omega(\mathscr{T}) + \eta\right\} \leqslant 2e^{-\frac{\eta^2}{8\sigma^2(\mathscr{T})}}.$$

Also, $Z_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} = Y_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} - \beta_{s,m}\|\boldsymbol{u}\|_{\ell_2}$ implies that

$$\mathbb{P}\left\{\bigcup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s,\ \boldsymbol{u}\in\mathscr{T},\boldsymbol{v}\in\mathbb{S}^{m-s-1}}\left[Y_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} > \beta_{s,m}\|\boldsymbol{u}\|_{\ell_2} + \omega(\mathscr{T}) + \eta\right]\right\} \leqslant 2e^{-\frac{\eta^2}{8\sigma^2(\mathscr{T})}}.$$

As we noted that the two Gaussian processes $X_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}}$ and $Y_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}}$ have the three assumptions of Slepian's inequality, we can use Slepian's second inequality (A2) with $\eta_{\boldsymbol{u},\boldsymbol{v}} = \beta_{s,m}\|\boldsymbol{u}\|_{\ell_2} + \eta + \omega(\mathscr{T})$. This implies that

$$\begin{aligned}
&\mathbb{P}\left\{\bigcup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s,\ \boldsymbol{u}\in\mathscr{T},\boldsymbol{v}\in\mathbb{S}^{m-s-1}}\left[X_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} > \beta_{s,m}\|\boldsymbol{u}\|_{\ell_2} + \omega(\mathscr{T}) + \eta\right]\right\}\\
&\quad\leqslant \mathbb{P}\left\{\bigcup_{|\mathscr{S}|=s,\ \boldsymbol{u}\in\mathscr{T},\boldsymbol{v}\in\mathbb{S}^{m-s-1}}\left[Y_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} > \beta_{s,m}\|\boldsymbol{u}\|_{\ell_2} + \omega(\mathscr{T}) + \eta\right]\right\}\\
&\quad\leqslant 2e^{-\frac{\eta^2}{8\sigma^2(\mathscr{T})}}.
\end{aligned}$$

Using the fact that $\left\|\boldsymbol{A}_{\mathscr{S}^c}\boldsymbol{u}\right\|_{\ell_2} = \max_{\boldsymbol{v}\in\mathbb{S}^{m-s-1}}\boldsymbol{v}^*\boldsymbol{I}_{\mathscr{S}^c}\boldsymbol{A}\boldsymbol{u} = \max_{\boldsymbol{v}\in\mathbb{S}^{m-s-1}}\boldsymbol{X}_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}}$, So

$$\begin{aligned}
&\mathbb{P}\left\{\sup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s}\left\|\boldsymbol{A}_{\mathscr{S}^c}\boldsymbol{u}\right\|_{\ell_2} \geqslant \beta_{s,m}\|\boldsymbol{u}\|_{\ell_2} + \omega(\mathscr{T}) + \eta\right\}\\
&\quad= \mathbb{P}\left\{\bigcup_{|\mathscr{S}|=s,\ \boldsymbol{u}\in\mathscr{T}}\max_{\boldsymbol{v}\in\mathbb{S}^{m-s-1}}\boldsymbol{X}_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} > \|\boldsymbol{u}\|_{\ell_2}\beta_{s,m} + \omega(\mathscr{T}) + \eta\right\}\\
&\quad= \mathbb{P}\left\{\bigcup_{|\mathscr{S}|=s,\ \boldsymbol{u}\in\mathscr{T},\ \boldsymbol{v}\in\mathbb{S}^{m-s-1}}\left[X_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} > \beta_{s,m}\|\boldsymbol{u}\|_{\ell_2} + \omega(\mathscr{T}) + \eta\right]\right\}
\end{aligned}$$

concludes the proof.

Next, we turn our attention to proving (5.6). To this aim we begin by stating a lemma due to Gordon [5].

LEMMA A.2 [Gordon's inequality] Let (X_{ij}) and (Y_{ut}), $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m$, be Gaussian random vectors. Assume that we have the following inequalities for all choices of indices:

$$\begin{aligned}
\mathbb{E}[X_{i,j}X_{i,k}] &\leqslant \mathbb{E}[Y_{i,j}Y_{i,k}] \quad \text{for all } i,j,k,\\
\mathbb{E}[X_{i,j}X_{\ell,k}] &\geqslant \mathbb{E}[Y_{i,j}Y_{\ell,k}] \quad \text{for all } i \neq \ell \text{ and } j,k,\\
\mathbb{E}[X_{i,j}^2] &= \mathbb{E}[Y_{i,j}^2] \qquad \text{for all } i,j.
\end{aligned} \tag{A.9}$$

Then, for all real numbers $\lambda_{i,j}$,

$$\mathbb{P}\left\{\bigcap_{i\leqslant n}\bigcup_{j\leqslant m}[X_{ij}\geqslant\lambda_{ij}]\right\}\geqslant\mathbb{P}\left\{\bigcap_{i\leqslant n}\bigcup_{j\leqslant m}[Y_{ij}\geqslant\lambda_{ij}]\right\}.$$

Consequently,

$$\mathbb{E}\left[\min_{i\leqslant n}\max_{j\leqslant m}Y_{ij}\right]\leqslant\mathbb{E}\left[\min_{i\leqslant n}\max_{j\leqslant m}X_{ij}\right].$$

Define $\boldsymbol{I}_{\mathscr{S}^c}\in\mathbb{R}^{(m-s)\times n}$ as the part of the identity matrix that keeps the rows indexed by $\mathscr{S}^c$. For $\boldsymbol{u}\in\mathscr{T}$ and $\boldsymbol{v}\in\mathbb{S}^{m-s-1}=\{\boldsymbol{v}\in\mathbb{R}^{m-s};\ \ \|\boldsymbol{v}\|_{\ell_2}=1\}$, we define two Gaussian processes

$$X_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}}=\boldsymbol{v}^*\boldsymbol{I}_{\mathscr{S}^c}\boldsymbol{A}\boldsymbol{u}+\|\boldsymbol{u}\|_{\ell_2}\,g\quad\text{and}\quad Y_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}}=\|\boldsymbol{u}\|_{\ell_2}\,\boldsymbol{v}^*\boldsymbol{I}_{\mathscr{S}^c}\boldsymbol{a}+\boldsymbol{g}^*\boldsymbol{u}.$$

Here, $\boldsymbol{a}\in\mathbb{R}^m$ is distributed as $\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_m)$, $\boldsymbol{g}\in\mathbb{R}^n$ is distributed as $\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_n)$, and $g\in\mathbb{R}$ is distributed as $\mathcal{N}(0,1)$. The next few steps are essentially identical to the proof of [5, Lemma 3.1] with the text directly borrowed. We mention this part of the argument for the sake of completeness and also to ensure that proper modifications are applied when necessary. Note that

$$\begin{aligned}
&\mathbb{E}\left[X_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}}X_{(\tilde{\boldsymbol{u}},\tilde{\mathscr{S}}),\tilde{\boldsymbol{v}}}\right]-\mathbb{E}\left[Y_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}}Y_{(\tilde{\boldsymbol{u}},\tilde{\mathscr{S}}),\tilde{\boldsymbol{v}}}\right]\\
&\quad=\langle\boldsymbol{u},\tilde{\boldsymbol{u}}\rangle\langle\boldsymbol{I}_{\mathscr{S}^c}^T\boldsymbol{v},\boldsymbol{I}_{\tilde{\mathscr{S}}^c}^T\tilde{\boldsymbol{v}}\rangle+\|\boldsymbol{u}\|_{\ell_2}\|\tilde{\boldsymbol{u}}\|_{\ell_2}-\|\boldsymbol{u}\|_{\ell_2}\|\tilde{\boldsymbol{u}}\|_{\ell_2}\langle\boldsymbol{I}_{\mathscr{S}^c}^T\boldsymbol{v},\boldsymbol{I}_{\tilde{\mathscr{S}}^c}^T\tilde{\boldsymbol{v}}\rangle-\langle\boldsymbol{u},\tilde{\boldsymbol{u}}\rangle\\
&\quad=\left(\|\boldsymbol{u}\|_{\ell_2}\|\tilde{\boldsymbol{u}}\|_{\ell_2}-\langle\boldsymbol{u},\tilde{\boldsymbol{u}}\rangle\right)\left(1-\langle\boldsymbol{I}_{\mathscr{S}^c}^T\boldsymbol{v},\boldsymbol{I}_{\tilde{\mathscr{S}}^c}^T\tilde{\boldsymbol{v}}\rangle\right)\\
&\quad\geqslant 0
\end{aligned}$$

and equal to zero if $(\boldsymbol{u},\mathscr{S})=(\tilde{\boldsymbol{u}},\tilde{\mathscr{S}})$ so that the first two inequalities in (A.9) hold. It is also trivial to check that

$$\mathbb{E}\left[X_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}}^2\right]=\mathbb{E}\left[Y_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}}^2\right].$$

Thus, all three inequalities in (A.9) trivially hold. Now note that for each $\boldsymbol{u}\in\mathscr{T}$ and $\mathscr{S}\subset\{1,2,\ldots,m\}$ obeying $|\mathscr{S}|=s$ the set

$$\bigcup_{\boldsymbol{v}\in\mathbb{S}^{m-s-1}}\left[X_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}}\geqslant\lambda_{\boldsymbol{u},\mathscr{S}}\right]=\left[\,\left\|\boldsymbol{A}_{\mathscr{S}^c}\boldsymbol{u}\right\|_{\ell_2}+g\,\|\boldsymbol{u}\|_{\ell_2}\geqslant\lambda_{\boldsymbol{u},\mathscr{S}}\right]$$

is closed in the probability space $\{\mathbb{R}^{mn+1},\mathbb{P}\}$, where $\mathbb{P}$ is the canonical Gaussian measure of $\mathbb{R}^{mn+1}$. Hence,

$$\bigcap_{|\mathscr{S}|=s}\bigcap_{\boldsymbol{u}\in\mathscr{T}}\bigcup_{\boldsymbol{v}\in\mathbb{S}^{m-s-1}}\left[X_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}}\geqslant\lambda_{\boldsymbol{u},\mathscr{S}}\right]$$

is closed. The same is true about the corresponding expression with $Y_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}}$. By Lemma A.2 above, for each finite set $\{(\boldsymbol{u}_i,\mathscr{S}_i)\}_1^N\subset\mathscr{T}\times\{1,2,\ldots,m\}$, we have

$$\mathbb{P}\left\{\bigcap_{i=1}^N\bigcup_{\boldsymbol{v}\in\mathbb{S}^{m-s-1}}\left[X_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}}\geqslant\lambda_{\boldsymbol{u},\mathscr{S}}\right]\right\}\geqslant\mathbb{P}\left\{\bigcap_{i=1}^N\bigcup_{\boldsymbol{v}\in\mathbb{S}^{m-s-1}}\left[Y_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}}\geqslant\lambda_{\boldsymbol{u},\mathscr{S}}\right]\right\}$$

and so, ordering the collection of finite subsets of $\mathscr{T} \times \{1, 2, \ldots, m\}$ (denoted by $\mathscr{F}$) by inclusion, we obtain that the limits exist and satisfy the inequality

$$\lim_{\mathscr{F}} \mathbb{P}\left\{\bigcap_{i=1}^{N} \bigcup_{\boldsymbol{v}\in\mathbb{S}^{m-s-1}} \left[X_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} \geqslant \lambda_{\boldsymbol{u},\mathscr{S}}\right]\right\} \geqslant \lim_{\mathscr{F}} \mathbb{P}\left\{\bigcap_{i=1}^{N} \bigcup_{\boldsymbol{v}\in\mathbb{S}^{m-s-1}} \left[Y_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} \geqslant \lambda_{\boldsymbol{u},\mathscr{S}}\right]\right\}.$$

Now using the fact that the sets

$$\bigcap_{|\mathscr{S}|=s} \bigcap_{\boldsymbol{u}\in\mathscr{T}} \bigcup_{\boldsymbol{v}\in\mathbb{S}^{m-s-1}} \left[X_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} \geqslant \lambda_{\boldsymbol{u},\mathscr{S}}\right] \quad \text{and} \quad \bigcap_{|\mathscr{S}|=s} \bigcap_{\boldsymbol{u}\in\mathscr{T}} \bigcup_{\boldsymbol{v}\in\mathbb{S}^{m-s-1}} \left[Y_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} \geqslant \lambda_{\boldsymbol{u},\mathscr{S}}\right]$$

are closed and $\mathbb{P}$ is a regular measure, it follows easily that the two respective limits over $\mathscr{F}$ are equal to and satisfy the inequality

$$\mathbb{P}\left\{\bigcap_{|\mathscr{S}|=s} \bigcap_{\boldsymbol{u}\in\mathscr{T}} \bigcup_{\boldsymbol{v}\in\mathbb{S}^{m-s-1}} \left[X_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} \geqslant \lambda_{\boldsymbol{u},\mathscr{S}}\right]\right\} \geqslant \mathbb{P}\left\{\bigcap_{|\mathscr{S}|=s} \bigcap_{\boldsymbol{u}\in\mathscr{T}} \bigcup_{\boldsymbol{v}\in\mathbb{S}^{m-s-1}} \left[Y_{(\boldsymbol{u},\mathscr{S}),\boldsymbol{v}} \geqslant \lambda_{\boldsymbol{u},\mathscr{S}}\right]\right\}.$$

This immediately implies that

$$\mathbb{P}\left\{\bigcap_{|\mathscr{S}|=s} \bigcap_{\boldsymbol{u}\in\mathscr{T}} \left[\left\|\boldsymbol{A}_{\mathscr{S}^c}\boldsymbol{u}\right\|_{\ell_2} + g\left\|\boldsymbol{u}\right\|_{\ell_2} \geqslant \lambda_{\boldsymbol{u},\mathscr{S}}\right]\right\} \geqslant \mathbb{P}\left\{\bigcap_{|\mathscr{S}|=s} \bigcap_{\boldsymbol{u}\in\mathscr{T}} \left[\left\|\boldsymbol{u}\right\|_{\ell_2}\left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2} + \boldsymbol{g}^*\boldsymbol{u} \geqslant \lambda_{\boldsymbol{u},\mathscr{S}}\right]\right\}.$$

Now setting

$$\lambda_{\boldsymbol{u},\mathscr{S}} = \alpha_{s,m}\left\|\boldsymbol{u}\right\|_{\ell_2} - (\omega(\mathscr{T}) + \eta),$$

we conclude that

$$\begin{aligned}
&\mathbb{P}\left\{\bigcap_{|\mathscr{S}|=s} \bigcap_{\boldsymbol{u}\in\mathscr{T}} \left[\left\|\boldsymbol{A}_{\mathscr{S}^c}\boldsymbol{u}\right\|_{\ell_2} + g\left\|\boldsymbol{u}\right\|_{\ell_2} \geqslant \alpha_{s,m}\left\|\boldsymbol{u}\right\|_{\ell_2} - (\omega(\mathscr{T}) + \eta)\right]\right\} \\
&\quad \geq \mathbb{P}\left\{\bigcap_{|\mathscr{S}|=s} \bigcap_{\boldsymbol{u}\in\mathscr{T}} \left[\left\|\boldsymbol{u}\right\|_{\ell_2}\left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2} + \boldsymbol{g}^*\boldsymbol{u} \geqslant \alpha_{s,m}\left\|\boldsymbol{u}\right\|_{\ell_2} - (\omega(\mathscr{T}) + \eta)\right]\right\} \\
&\quad = \mathbb{P}\left\{\bigcap_{\boldsymbol{u}\in\mathscr{T}} \left[\left\|\boldsymbol{u}\right\|_{\ell_2} \inf_{|\mathscr{S}|=s} \left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2} + \boldsymbol{g}^*\boldsymbol{u} \geqslant \alpha_{s,m}\left\|\boldsymbol{u}\right\|_{\ell_2} - (\omega(\mathscr{T}) + \eta)\right]\right\}. \qquad \text{(A.10)}
\end{aligned}$$

Since taking infimum over a set is equivalent to taking intersection over all elements of that set, we can write

$$\begin{aligned}\mathbb{P}\left\{\bigcap_{\boldsymbol{u}\in\mathscr{T}}\left[\boldsymbol{g}^*\boldsymbol{u}\geqslant-\left(\omega(\mathscr{T})+\frac{\eta}{2}\right)\right]\right\}&=\mathbb{P}\left\{\inf_{\boldsymbol{u}\in\mathscr{T}}\boldsymbol{g}^*\boldsymbol{u}\geqslant-\left(\omega(\mathscr{T})+\frac{\eta}{2}\right)\right\}\\&=\mathbb{P}\left\{-\sup_{\boldsymbol{u}\in\mathscr{T}}-\boldsymbol{g}^*\boldsymbol{u}\geqslant-\left(\omega(\mathscr{T})+\frac{\eta}{2}\right)\right\},\\&=\mathbb{P}\left\{\sup_{\boldsymbol{u}\in\mathscr{T}}-\boldsymbol{g}^*\boldsymbol{u}\leqslant\left(\omega(\mathscr{T})+\frac{\eta}{2}\right)\right\},\\&\geqslant 1-e^{-\frac{\eta^2}{8\sigma^2(\mathscr{T})}}.\end{aligned}\tag{A.11}$$

In the last inequality, we used the concentration of Lipschitz functions of Gaussians and the definition of Gaussian width.

Now define the function $g(\boldsymbol{x})=\inf_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s}\left\|\boldsymbol{x}_{\mathscr{S}^c}\right\|_{\ell_2}$ and let

$$\mathscr{S}_{\boldsymbol{x}}^c=\underset{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s}{\arg\min}\left\|\boldsymbol{x}_{\mathscr{S}^c}\right\|_{\ell_2}\quad\text{and}\quad\mathscr{S}_{\boldsymbol{y}}^c=\underset{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s}{\arg\min}\left\|\boldsymbol{y}_{\mathscr{S}^c}\right\|_{\ell_2}.$$

We claim that $g(\boldsymbol{x})$ is a Lipschitz function and then we can utilize the concentration of measure for Gaussian random variables. Without loss of generality, we assume $g(\boldsymbol{x})\geqslant g(\boldsymbol{y})$. Thus,

$$\begin{aligned}|g(\boldsymbol{x})-g(\boldsymbol{y})|=g(\boldsymbol{x})-g(\boldsymbol{y})&=\left\|\boldsymbol{x}_{\mathscr{S}_{\boldsymbol{x}}^c}\right\|_{\ell_2}-\left\|\boldsymbol{y}_{\mathscr{S}_{\boldsymbol{y}}^c}\right\|_{\ell_2}\\&\leqslant\left\|\boldsymbol{x}_{\mathscr{S}_{\boldsymbol{y}}^c}\right\|_{\ell_2}-\left\|\boldsymbol{y}_{\mathscr{S}_{\boldsymbol{y}}^c}\right\|_{\ell_2}\leqslant\left\|(\boldsymbol{x}-\boldsymbol{y})_{\mathscr{S}_{\boldsymbol{y}}^c}\right\|_{\ell_2}\leqslant\|\boldsymbol{x}-\boldsymbol{y}\|_{\ell_2}.\end{aligned}$$

Hence, $g(\boldsymbol{a})$ is a Lipschitz function of a Gaussian random variable. Thus, the random variable $Z:=g(\boldsymbol{a})$ obeys

$$\mathrm{Var}(Z)\leqslant 1\quad\Rightarrow\quad\mathbb{E}[Z^2]-(\mathbb{E}[Z])^2\leqslant 1\quad\Rightarrow\quad\mathbb{E}[Z]\geqslant\sqrt{\mathbb{E}[Z^2]-1}.$$

Therefore,

$$\begin{aligned}\mathbb{E}\left[\inf_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s}\left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2}\right]&\geqslant\sqrt{\mathbb{E}\left[\inf_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s}\left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2}^2\right]-1}\\&\geqslant\sqrt{\mathbb{E}\left[\|\boldsymbol{a}\|_{\ell_2}^2-\sup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s}\left\|\boldsymbol{a}_{\mathscr{S}}\right\|_{\ell_2}^2\right]-1}\\&=\sqrt{m-1-\mathbb{E}\left[\sup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s}\left\|\boldsymbol{a}_{\mathscr{S}}\right\|_{\ell_2}^2\right]}\\&\geqslant\sqrt{m-2-\mathbb{E}\left[\sup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s}\left\|\boldsymbol{a}_{\mathscr{S}}\right\|_{\ell_2}\right]^2}.\end{aligned}\tag{A.12}$$

In the last line, we used the fact that $\sup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s} \left\|\boldsymbol{a}_{\mathscr{S}}\right\|_{\ell_2}$ is a Lipschitz function of $\boldsymbol{a}$ and therefore the random variable $X := \sup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s} \left\|\boldsymbol{a}_{\mathscr{S}}\right\|_{\ell_2}$ obeys

$$\mathrm{Var}(X) \leqslant 1 \quad \Rightarrow \quad \mathbb{E}[X^2] \leqslant (\mathbb{E}[X])^2 + 1.$$

We now wish to prove that

$$\mathbb{E}\left[\sup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s} \left\|\boldsymbol{a}_{\mathscr{S}}\right\|_{\ell_2}\right] \leqslant \sqrt{5s\log\left(\frac{em}{s}\right)}. \tag{A.13}$$

To this aim first note that using (A.4) with changing $m-s$ to s and $\mathscr{S}^c$ to $\mathscr{S}$, we have

$$\mathbb{P}\left\{\sup_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s} \left\|\boldsymbol{a}_{\mathscr{S}}\right\|_{\ell_2} \geqslant \eta + \sqrt{s} + \sqrt{2s\log\left(\frac{em}{s}\right)}\right\} \leqslant e^{\frac{-\eta^2}{2}}.$$

To bound the expected value, we use the tail bound above together with the fact that $s \geqslant 1$ ($s=0$ is trivial) to conclude that

$$\begin{aligned}
\mathbb{E}[X] &= \int_0^{+\infty} \mathbb{P}\{X > t\}\mathrm{d}t \\
&= \int_0^{\sqrt{s}+\sqrt{2s\log\left(\frac{em}{s}\right)}} \mathbb{P}\{X > t\}\mathrm{d}t + \int_{\sqrt{s}+\sqrt{2(s)\log\left(\frac{em}{s}\right)}}^{+\infty} \mathbb{P}\{X > t\}\mathrm{d}t \\
&\leqslant \sqrt{s} + \sqrt{2s\log\left(\frac{em}{s}\right)} + \int_{\sqrt{s}+\sqrt{2s\log\left(\frac{em}{s}\right)}}^{+\infty} e^{\frac{-\eta^2}{2}}\mathrm{d}t \\
&\leqslant \sqrt{s} + \sqrt{2s\log\left(\frac{em}{s}\right)} + \int_0^{+\infty} e^{\frac{-\eta^2}{2}}\mathrm{d}t \\
&\leqslant \sqrt{s} + \sqrt{2s\log\left(\frac{em}{s}\right)} + \sqrt{\frac{\pi}{2}} \\
&\leqslant \sqrt{s} + \sqrt{2s\log\left(\frac{em}{s}\right)} + \sqrt{\frac{\pi s}{2}} \\
&\leqslant \sqrt{5s\log\left(\frac{em}{s}\right)}
\end{aligned}$$

concluding the proof of (A.13).

Combining (A.12) with (A.13), we arrive at

$$\mathbb{E}\left[\inf_{\mathscr{S}\subset\{1,2,\ldots,m\},\ |\mathscr{S}|=s} \left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2}\right] \geqslant \sqrt{m-2-5s\log\left(\frac{em}{s}\right)} := \alpha_{s,m}. \tag{A.14}$$

As mentioned earlier, $g(\boldsymbol{a})$ is Lipschitz function of $\boldsymbol{a}$. Thus, by concentration of Lipschitz functions of Gaussians, we have

$$\mathbb{P}\left\{g(\boldsymbol{a}) \geqslant \mathbb{E}[g(\boldsymbol{a})] - \frac{\eta}{2\sigma(\mathscr{T})}\right\} \geqslant 1 - e^{-\frac{\eta^2}{8\sigma^2(\mathscr{T})}}. \tag{A.15}$$

Using the fact that $\frac{\|\boldsymbol{u}\|_{\ell_2}}{\sigma(\mathscr{T})} \leqslant 1$ and together with (A.12) as well as (A.15), we can deduce that

$$\mathbb{P}\left\{\bigcap_{\boldsymbol{u}\in\mathscr{T}}\left[\|\boldsymbol{u}\|_{\ell_2}\inf_{|\mathscr{S}|=s}\left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2} \geqslant \|\boldsymbol{u}\|_{\ell_2}\alpha_{s,m} - \frac{\eta}{2}\right]\right\} \tag{A.16}$$

$$\geqslant \mathbb{P}\left\{\bigcap_{\boldsymbol{u}\in\mathscr{T}}\left[\|\boldsymbol{u}\|_{\ell_2}\inf_{|\mathscr{S}|=s}\left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2} \geqslant \|\boldsymbol{u}\|_{\ell_2}\mathbb{E}\left[\inf_{|\mathscr{S}|=s}\left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2}\right] - \frac{\eta}{2}\right]\right\}$$

$$\geqslant \mathbb{P}\left\{\bigcap_{\boldsymbol{u}\in\mathscr{T}}\left[\|\boldsymbol{u}\|_{\ell_2}\, g(\boldsymbol{a}) \geqslant \|\boldsymbol{u}\|_{\ell_2}\,\mathbb{E}[g(\boldsymbol{a})] - \|\boldsymbol{u}\|_{\ell_2}\,\frac{\eta}{2\sigma(\mathscr{T})}\right]\right\}$$

$$\geqslant 1 - e^{-\frac{\eta^2}{8\sigma^2(\mathscr{T})}}. \tag{A.17}$$

So far we have obtained lower bounds on the probability of sets $\left\{\bigcap_{\boldsymbol{u}\in\mathscr{T}}\left[\boldsymbol{g}^*\boldsymbol{u} \geqslant -\left(\omega(\mathscr{T}) + \frac{\eta}{2}\right)\right]\right\}$ and $\left\{\bigcap_{\boldsymbol{u}\in\mathscr{T}}\left[\|\boldsymbol{u}\|_{\ell_2}\inf_{|\mathscr{S}|=s}\left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2} \geqslant \|\boldsymbol{u}\|_{\ell_2}\alpha_{s,m} - \frac{\eta}{2}\right]\right\}$. In the following, we aim to employ these two lower bounds. Note that if

$$\inf_{\boldsymbol{u}\in\mathscr{T}}\left(\|\boldsymbol{u}\|_{\ell_2}\left(\inf_{|\mathscr{S}|=s}\left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2} - \alpha_{s,m}\right) + \boldsymbol{g}^*\boldsymbol{u}\right) < -\left(\omega(\mathscr{T}) + \eta\right),$$

then we have either $\inf_{\boldsymbol{u}\in\mathscr{T}}\|\boldsymbol{u}\|_{\ell_2}\left(\inf_{|\mathscr{S}|=s}\left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2} - \alpha_{s,m}\right) < -\frac{\eta}{2}$ or $\inf_{\boldsymbol{u}\in\mathscr{T}}(\boldsymbol{g}^*\boldsymbol{u}) < -\left(\omega(\mathscr{T}) + \frac{\eta}{2}\right)$. This implies that

$$\left\{\boldsymbol{a},\boldsymbol{g}:\ \inf_{\boldsymbol{u}\in\mathscr{T}}\left(\|\boldsymbol{u}\|_{\ell_2}\left(\inf_{|\mathscr{S}|=s}\left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2} - \alpha_{s,m}\right) + \boldsymbol{g}^*\boldsymbol{u}\right) < -\left(\omega(\mathscr{T}) + \eta\right)\right\},$$

is a subset of

$$\left\{\boldsymbol{a},\boldsymbol{g}:\ \inf_{\boldsymbol{u}\in\mathscr{T}}\|\boldsymbol{u}\|_{\ell_2}\left(\inf_{|\mathscr{S}|=s}\left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2} - \alpha_{s,m}\right) < -\frac{\eta}{2}\right\} \bigcup \left\{\boldsymbol{a},\boldsymbol{g}:\ \inf_{\boldsymbol{u}\in\mathscr{T}}\left(\boldsymbol{g}^*\boldsymbol{u}\right) < -\left(\omega(\mathscr{T}) + \frac{\eta}{2}\right)\right\}.$$

The latter is equivalent to

$$\bigcup_{\boldsymbol{u}\in\mathscr{T}}\left[\,\|\boldsymbol{u}\|_{\ell_2}\inf_{|\mathscr{S}|=s}\left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2} + \boldsymbol{g}^*\boldsymbol{u} < \alpha_{s,m}\|\boldsymbol{u}\|_{\ell_2} - \left(\omega(\mathscr{T}) + \eta\right)\right]$$

$$\subset \left(\bigcup_{\boldsymbol{u}\in\mathscr{T}}\left[\|\boldsymbol{u}\|_{\ell_2}\inf_{|\mathscr{S}|=s}\left\|\boldsymbol{a}_{\mathscr{S}^c}\right\|_{\ell_2} < \|\boldsymbol{u}\|_{\ell_2}\alpha_{s,m} - \frac{\eta}{2}\right]\right) \bigcup \left(\bigcup_{\boldsymbol{u}\in\mathscr{T}}\left[\boldsymbol{g}^*\boldsymbol{u} < -\left(\omega(\mathscr{T}) + \frac{\eta}{2}\right)\right]\right).$$

Considering the probability of these sets, we have

$$\mathbb{P}\Big\{\bigcup_{\boldsymbol{u}\in\mathscr{T}}\big[\,\|\boldsymbol{u}\|_{\ell_2}\inf_{|\mathscr{S}|=s}\|\boldsymbol{a}_{\mathscr{S}^c}\|_{\ell_2}+\boldsymbol{g}^*\boldsymbol{u}<\alpha_{s,m}\|\boldsymbol{u}\|_{\ell_2}-(\omega(\mathscr{T})+\eta)\,\big]\Big\}$$
$$\leqslant\mathbb{P}\Big\{\bigcup_{\boldsymbol{u}\in\mathscr{T}}\Big[\,\|\boldsymbol{u}\|_{\ell_2}\inf_{|\mathscr{S}|=s}\|\boldsymbol{a}_{\mathscr{S}^c}\|_{\ell_2}<\|\boldsymbol{u}\|_{\ell_2}\alpha_{s,m}-\frac{\eta}{2}\Big]\Big\}+\mathbb{P}\Big\{\bigcup_{\boldsymbol{u}\in\mathscr{T}}\Big[\boldsymbol{g}^*\boldsymbol{u}<-\Big(\omega(\mathscr{T})+\frac{\eta}{2}\Big)\,\Big]\Big\}.$$

Taking complements of both sides and using the bounds from (A.11) and (A.16), we conclude that

$$\mathbb{P}\Big\{\bigcap_{\boldsymbol{u}\in\mathscr{T}}\big[\,\|\boldsymbol{u}\|_{\ell_2}\inf_{|\mathscr{S}|=s}\|\boldsymbol{a}_{\mathscr{S}^c}\|_{\ell_2}+\boldsymbol{g}^*\boldsymbol{u}\geqslant\alpha_{s,m}\|\boldsymbol{u}\|_{\ell_2}-(\omega(\mathscr{T})+\eta)\,\big]\Big\}$$
$$\geqslant\mathbb{P}\Big\{\bigcap_{\boldsymbol{u}\in\mathscr{T}}\Big[\,\|\boldsymbol{u}\|_{\ell_2}\inf_{|\mathscr{S}|=s}\|\boldsymbol{a}_{\mathscr{S}^c}\|_{\ell_2}\geqslant\|\boldsymbol{u}\|_{\ell_2}\alpha_{s,m}-\frac{\eta}{2}\Big]\Big\}+\mathbb{P}\Big\{\bigcap_{\boldsymbol{u}\in\mathscr{T}}\Big[\boldsymbol{g}^*\boldsymbol{u}\geqslant-\Big(\omega(\mathscr{T})+\frac{\eta}{2}\Big)\,\Big]\Big\}-1$$
$$\geqslant 1-2e^{-\frac{\eta^2}{8\sigma^2(\mathscr{T})}}. \tag{A.18}$$

The latter inequality together with (A.10) implies that

$$\mathbb{P}\Big\{\inf_{|\mathscr{S}|=s}\inf_{\boldsymbol{u}\in\mathscr{T}}\big(\|\boldsymbol{A}_{\mathscr{S}^c}\boldsymbol{u}\|_{\ell_2}+g\,\|\boldsymbol{u}\|_{\ell_2}-\alpha_{s,m}\|\boldsymbol{u}\|_{\ell_2}\big)\geqslant-(\omega(\mathscr{T})+\eta)\Big\}$$
$$=\mathbb{P}\Big\{\bigcap_{|\mathscr{S}|=s}\bigcap_{\boldsymbol{u}\in\mathscr{T}}\big[\,\|\boldsymbol{A}_{\mathscr{S}^c}\boldsymbol{u}\|_{\ell_2}+g\,\|\boldsymbol{u}\|_{\ell_2}-\alpha_{s,m}\|\boldsymbol{u}\|_{\ell_2}\geqslant-(\omega(\mathscr{T})+\eta)\,\big]\Big\}$$
$$\geqslant 1-2e^{-\frac{\eta^2}{8\sigma^2(\mathscr{T})}}. \tag{A.19}$$

In order to find the relationship between the probability of the latter set with the probability of the set defined in (5.6), we define the following three probabilities:

$$p=\mathbb{P}\Big\{\inf_{|\mathscr{S}|=s}\inf_{\boldsymbol{u}\in\mathscr{T}}\Big(\|\boldsymbol{A}_{\mathscr{S}^c}\boldsymbol{u}\|_{\ell_2}+g\,\|\boldsymbol{u}\|_{\ell_2}-\alpha_{s,m}\|\boldsymbol{u}\|_{\ell_2}\Big)\geqslant-(\omega(\mathscr{T})+\eta)\Big\},$$
$$p_-=\mathbb{P}\Big\{\inf_{|\mathscr{S}|=s}\inf_{\boldsymbol{u}\in\mathscr{T}}\Big(\|\boldsymbol{A}_{\mathscr{S}^c}\boldsymbol{u}\|_{\ell_2}+g\,\|\boldsymbol{u}\|_{\ell_2}-\alpha_{s,m}\|\boldsymbol{u}\|_{\ell_2}\Big)\geqslant-(\omega(\mathscr{T})+\eta)\;\Big|\;g\leqslant 0\Big\},$$
$$p_+=\mathbb{P}\Big\{\inf_{|\mathscr{S}|=s}\inf_{\boldsymbol{u}\in\mathscr{T}}\Big(\|\boldsymbol{A}_{\mathscr{S}^c}\boldsymbol{u}\|_{\ell_2}+g\,\|\boldsymbol{u}\|_{\ell_2}-\alpha_{s,m}\|\boldsymbol{u}\|_{\ell_2}\Big)\geqslant-(\omega(\mathscr{T})+\eta)\;\Big|\;g>0\Big\},$$
$$p_0=\mathbb{P}\Big\{\inf_{|\mathscr{S}|=s}\inf_{\boldsymbol{u}\in\mathscr{T}}\Big(\|\boldsymbol{A}_{\mathscr{S}^c}\boldsymbol{u}\|_{\ell_2}-\alpha_{s,m}\|\boldsymbol{u}\|_{\ell_2}\Big)\geqslant-(\omega(\mathscr{T})+\eta)\;\Big\}.$$

Now note that by the above definitions and the independence of $\boldsymbol{A}$ and g, we can conclude that

$$1\geqslant p_+\geqslant p_0\geqslant p_-. \tag{A.20}$$

By the law of total probability $p = \frac{p_- + p_+}{2}$. Now using the fact that $p_+ \leqslant 1$ together with (A.20), we can conclude that

$$1 - p = \frac{1 - p_-}{2} + \frac{1 - p_+}{2} \geqslant \frac{1 - p_-}{2} \geqslant \frac{1 - p_0}{2} \quad \Rightarrow \quad p_0 \geqslant 2p - 1.$$

The latter inequality together with (A.19) implies that

$$\mathbb{P}\left\{ \inf_{|\mathscr{S}|=s} \inf_{\boldsymbol{u} \in \mathscr{T}} \left(\left\| \boldsymbol{A}_{\mathscr{S}^c} \boldsymbol{u} \right\|_{\ell_2} + g \left\| \boldsymbol{u} \right\|_{\ell_2} - \alpha_{s,m} \left\| \boldsymbol{u} \right\|_{\ell_2} \right) \geqslant - \left(\omega(\mathscr{T}) + \eta \right) \right\} \geqslant 1 - 4e^{-\frac{\eta^2}{8\sigma^2(\mathscr{T})}},$$

concluding the proof of (5.6).