Maximally Recoverable Codes for Grid-like Topologies

Parikshit Gopalan
Microsoft Research
parik@microsoft.com

Guangda Hu
Princeton University
guangdah@cs.princeton.edu

Shubhangi Saraf
Rutgers University
shubhangi.saraf@gmail.com

Carol Wang
National University of Singapore
elecaro@nus.edu.sg

Sergey Yekhanin
Microsoft Research
yekhanin@microsoft.com

Abstract

The explosion in the volumes of data being stored online has resulted in distributed storage systems transitioning to erasure coding based schemes. Yet, the codes being deployed in practice are fairly short. In this work, we address what we view as the main coding theoretic barrier to deploying longer codes in storage: at large lengths, failures are not independent and correlated failures are inevitable. This motivates designing codes that allow quick data recovery even after large correlated failures, and which have efficient encoding and decoding.

We propose that code design for distributed storage be viewed as a two step process. The first step is choose a topology of the code, which incorporates knowledge about the correlated failures that need to be handled, and ensures local recovery from such failures. In the second step one specifies a code with the chosen topology by choosing coefficients from a finite field \( \mathbb{F}_q \). In this step, one tries to balance reliability (which is better over larger fields) with encoding and decoding efficiency (which is better over smaller fields).

This work initiates an in-depth study of this reliability/efficiency tradeoff. We consider the field-size needed for achieving maximal recoverability: the strongest reliability possible with a given topology. We propose a family of topologies called grid-like topologies which unify a number of topologies considered both in theory and practice, and prove the following results about codes for such topologies:

- The first super-polynomial lower bound on the field size needed for achieving maximal recoverability in a simple grid-like topology. To our knowledge, there was no super-linear lower bound known before, for any topology.

- A combinatorial characterization of erasure patterns correctable by Maximally Recoverable codes for a topology which corresponds to tensoring MDS codes with a parity check code. This topology is used in practice (for instance see [MLR+14]). We conjecture a similar characterization for Maximally Recoverable codes instantiating arbitrary tensor product topologies.

- A new asymptotically optimal family of Maximally Recoverable codes for a certain basic topology completing the line of work in [BHH13, GHJY14]
1 Introduction

The explosion in the volumes of data being stored online means that duplicating or triplicating data is not economically feasible. This has resulted in distributed storage systems employing erasure coding based schemes in order to ensure reliability with low storage overheads. Spurred by this, there has been an active line of research in coding theory focusing on distributed storage. Two main paradigms have emerged from this work: local reconstruction [GHSY12, SAP+13] and local regeneration [DGW+10], both focusing on the efficiency of the decoder in typical failure scenarios (which in storage means one or a few machines being unavailable, perhaps temporarily). The former focuses on the number of disk reads needed to handle such failures, the latter on the amount of communication. In the last few years, the theory around these codes has developed rapidly. There are constructions known that achieve optimality for various tradeoffs. A number of these codes have been deployed at scale in the real world [HSX+12, MLR+14].

Yet, the length of codes being used for data storage thus far is quite small: often in the low double digits. The coding-theoretic motivation for moving to larger lengths is obvious: coding at larger lengths allows better error-tolerance for a given overhead. There is also ample practical motivation, coming from the need to reduce storage costs. Increasingly, data stored in the cloud are geographically distributed across data centers, so that even if one location is offline for some time, data are still accessible. The simple solution of replication across data centers is expensive, and can nullify the gains from erasure coding within a data center. Finally, historical trends suggest that the transition to longer length codes should happen eventually. Thus it is important to understand what the current barriers to using longer codes are.

In this work, we address what we view as the main coding theoretic barrier to deploying longer codes in storage: at large lengths, the assumption that various nodes fail independently is just not true, correlated failures are inevitable. This motivates the task of designing codes that allow quick data recovery even after large correlated failures, and which have efficient encoding and decoding.

1.1 Codes with a topology

The coding theoretic challenges arising in distributed storage setting are very different from those encountered when codes are used for transmission, or even storage on a single device. There are two main reasons behind it:

- **Correlated failures**: In distributed storage, at large lengths, one cannot assume that individual codeword coordinates fail independently. One has to explicitly deal with large correlated failures, which might have different sources e.g. a rack failure, a datacenter failure, a simultaneous upgrade applied to a large group of machines, or failure of a power source shared by multiple machines. The structure of such correlated failures varies with deployment but is typically known at the code design stage, and can be incorporated in the code layout.

- **The need for locality**: Locality addresses the challenge of efficiently serving requests for unavailable data and maintaining an erasure encoded representation. In particular, when a node or a correlated group of nodes fails, one has to be able to quickly reconstruct every lost node in order to keep the data readily available for the users and to maintain the same level of redundancy in the system. We say that a certain packet has locality $r$ if it can be recovered from accessing only $r$ other packets (think of $r$ as being much less than the codeword length). We would like to ensure locality for typical failure scenarios. At short lengths with
independent failures, a single or a few failures might be a reasonable model for what is typical. But at longer lengths, we would like locality after correlated failures (like a data center being offline), which might mean that a constant fraction of machines is unavailable.

As a result, in designing codes for distributed storage one tries to incorporate knowledge about correlated failures in the design, in a way that guarantees efficient handling of such failures. The kinds of code construction problems that arise from this are different from those in classical coding theory, but we feel they are ripe for theoretical analysis. To facilitate this, we propose viewing the design of erasure codes for distributed storage as a two step process, where we intentionally separate out incorporating real-world knowledge about correlated failure patterns from code specification, which is very much within of realm of coding theoretic techniques.

1. **Picking a topology:** The first step is to determine the topology of the code, driven by the particular collection of correlated failures that need to be handled. Informally, one can think of a topology as specifying the supports for the parity check equations, but not the coefficients (or even which field they lie in). The topology specifies the number of redundant symbols and the data symbols that each of them depends on. This can be used to ensure the existence of short linear dependencies between specific codeword coordinates, so that the desired locality property holds for the correlated failure patterns of interest. This is the step which incorporates real-world knowledge about likely correlated failures into the design.

2. **Specifying coefficients:** In the second step one explicitly specifies a code with the chosen topology. We choose coefficients from a finite field $\mathbb{F}_q$, which fixes the redundant symbols as explicit $\mathbb{F}_q$-linear combinations of data symbols respecting the dependency constraints from the previous stage. This step typically utilizes tools from classical coding theory, but the objectives are different:

   - **Optimizing encoding/decoding efficiency:** Encoding a linear code and decoding it from erasures involve matrix vector multiplication and linear equation solving respectively. Both of these require performing numerous finite field arithmetic operations. Having small finite fields results in faster encoding and decoding and thus improves the overall throughput of the system [PGM13, Section 2]. In theory, field sizes which scale polynomially in the codeword length are desirable. Coefficient sizes of a few bytes are preferred in practice.

   - **Maximizing reliability:** Worst-case distance or the number of random failures tolerated are unsatisfactory reliability measures for codes with a prescribed topology. The notion of maximal recoverability first proposed by [CHL07] and generalized by [GIJRY14] provides a conceptually simple answer to the question *what is the best code for a given topology?*. Once we fix a topology and a set of erasures, decoding reduces to solving a system of linear equations. Maximal recoverability requires that the code corrects every failure pattern which is permissible by linear algebra, given the topology. Equivalently, a Maximally Recoverable (MR) code corrects every erasure pattern that is correctable for some fixing of coefficients in the same topology.

Reed Solomon codes are maximally recoverable codes for the trivial topology, and they have a linear field size (which is known to be optimal up to constant factors). However, this seems
to be an exception rather than the rule. Existing results suggest that it is hard to achieve these goals simultaneously, even in topologies are only slightly more complex than Reed-Solomon (see for instance [GHJY14]). For arbitrary topologies, random codes are maximally recoverable but over fields of exponential size, and often nothing better is known.

This points at a possible tradeoff between these two requirements. This tradeoff is the main subject of our work. It may be the case that in some topologies, the field-sizes required to achieve maximal recoverability are prohibitively large, so one needs to pick a different point on the tradeoff curve. A starting point for exploring this tradeoff is to understand the failure patterns that can be corrected by maximally recoverable codes for a topology, a problem that can again be challenging even in simple settings. Given this discussion, we propose the following questions as the natural main goals in the study of maximal recoverability.

For a given topology

- Determine the smallest field size over which MR codes exist.
- Characterize the failure patterns that can be corrected by MR codes.
- Find explicit constructions of MR codes over small fields.

In theory one could ask these questions about any topology, but the important topologies are simple ones which model how machines tend to be laid out within and across data centers. In this work, we propose a family of topologies called grid-like topologies which unify a number of topologies considered both in coding theory and practice. In short, codes with grid-like topologies can be viewed as tensor products of row and column codes, augmented with global parity check constraints. They provide a unified framework for MDS codes, tensor product codes, LRCs and more (see the discussion in Section 2).

We prove the following results about codes for grid-like topologies:

- The first super-polynomial lower bound on the field size needed for achieving maximal recoverability in any topology (in fact our bound applies to a very simple grid-like topology).
- A combinatorial characterization of erasure patterns correctable by Maximally Recoverable codes for a topology which corresponds to tensoring MDS codes with a parity check code. This topology is used in practice (for instance see Facebook’s f4 storage system [MLR+14]).
- A new asymptotically optimal family of Maximally Recoverable codes for a certain basic topology completing the line of work in [BHH13, GHJY14]

1.2 Related Work

The first family of codes with locality for applications in storage comes from [HCL07]. That paper also introduced the concept of maximal recoverability, in a restricted setting which does not allow for locality among parity check symbols. In this restricted setting, they gave a combinatorial characterization of correctable failure patterns via Hall’s theorem.

The approach of using a topology to ensure local recovery from correlated failures has been studied in the literature [BHH13, Bla13, GHJY14] and is used in practice [HSX+12, MLR+14]. The first definition of maximal recoverability for an arbitrary topology was given [GHJY14].

The work of [GHSY12] introduced a formal definition of locality, and focused on codes that guarantee locality for a single failure. For this simple setting, they were able to show that optimal
codes must have a certain natural topology. Maximally recoverable codes for that topology had been studied earlier in the work of [BHH13, Bla13], the best known general constructions are due to [GHJY14].

Discussion regarding the importance of using small finite fields in codes for storage can be found in [PGM13, Section 2].

The study of codes with locality and in particular maximally recoverable codes is distantly related to the study of Locally Decodable Codes (LDCs) [Yek12]. The key differences are as follows: LDCs can be viewed as codes where every symbol has low locality even after a constant-fraction of codeword coordinates are erased. The main challenge is to minimize the codeword length of these codes given the locality constraints. Instead, MR codes only provide locality after certain structured failures, the layout of which is known at the stage of code design and that are few in number. Codeword length is fixed by specifying the topology, and the key challenge is to minimize the field size while providing optimal erasure correction guarantees.

1.3 Outline

We start Section 2 with a formal definition of grid-like topologies. We explain why that definition indeed captures the needs that arise in distributed storage. In Section 3 we present formal statements of our three main theorems: the lower bound for alphabet size of MR codes, the combinatorial classification of erasure patterns correctable by MR codes, and an upper bound for the alphabet size of MR codes. Each of these results applies to a specific topology family. Remainder of the paper is dedicated to proofs. In Sections 4.1 and 4.2 we establish our alphabet size lower bound. In Section 5 we obtain the classification result. In Section 6 we give our new construction of maximally recoverable codes. Finally, in Section 7 we conclude the paper and discuss the key questions that remain open.

1.4 Some background and notation

We use the following standard mathematical notation:

- $[s] = \{1, \ldots, s\}$;
- Let $w \in \mathbb{F}^{n}$ be a vector. Let $\text{supp}(w) \subseteq [n]$ denote the set of non-zero coordinates of $w$.
- $[n, k, d]_q$ denotes a linear code (subspace) of dimension $k$, codeword length $n$, and distance $d$ over a field $\mathbb{F}_q$. We often write $[n, k, d]$ instead of $[n, k, d]_q$ when the particular choice of the field is not important.
- Let $C$ be an $[n, k, d]$ code and $S \subseteq [n]$, $|S| = k$. We say that $S$ is an information set if the restriction $C|_S = \mathbb{F}_q^k$.
- An $[n, k, d]$ code is called Maximum Distance Separable (MDS) if $d = n - k + 1$. MDS codes have many nice properties. In particular an $[n, k, d]$ code is MDS if and only if every subset of its $k$ coordinates is an information set. Alternatively, an $[n, k, d]$ code is MDS if and only if it corrects any collection of $(n - k)$ simultaneous erasures [MS77].
- Let $C_1$ be an $[n_1, k_1, d_1]$ code and $C_2$ be an $[n_2, k_2, d_2]$ code. The tensor product $C_1 \otimes C_2$ is an $[n_1n_2, k_1k_2, d_1d_2]$ code where the codewords of $C_1 \otimes C_2$ are matrices of size $n_1 \times n_2$, where
each column belongs to $C_1$ and each row belongs to $C_2$. If $U \subseteq [n_1]$ is an information set of $C_1$ and $V \subseteq [n_2]$ is an information set of $C_2$; then $U \times V$ is an information set of $C_1 \otimes C_2$, e.g., [MS77].

2 Grid-like topologies

We propose studying maximal recoverability for a simple class of topologies called grid-like topologies that unify and generalize many of the layouts that are used in practice [HSX+12, MLR+14]. Below is our key definition. We specify topologies via dual constraints. This way of defining topologies simplifies the proofs, however it might not be immediately clear that topologies defined like that indeed capture the needs that arise in distributed storage. We explain the connection in Proposition 4.

**Definition 1.** Let $m \leq n$ be integers. Consider an $m \times n$ array of symbols $\{x_{ij}\}_{i \in [m], j \in [n]}$ over a field $F$ of characteristic 2. Let $0 \leq a \leq m - 1, 0 \leq b \leq n - 1, \text{ and } 0 \leq h \leq (m - a)(n - b) - 1$. Let $T_{m \times n}(a, b, h)$ denote the topology where there are $a$ parity check equations per column, $b$ parity check equations per row, and $h$ global parity check equations that depend on all symbols. A code with this topology is specified by field elements $\{\alpha_i^{(k)}\}_{i \in [m], k \in [a]}$, $\{\beta_j^{(k)}\}_{j \in [n], k \in [b]}$ and $\{\gamma_{ij}^{(k)}\}_{i \in [m], j \in [n], k \in [h]}$.

1. Each column $j \in [n]$ satisfies the constraints
   \[ \sum_{i=1}^{m} \alpha_i^{(k)} x_{ij} = 0 \quad \forall k \in [a]. \]  
   (1)

2. Each row $i \in [m]$ satisfies the constraints:
   \[ \sum_{j=1}^{n} \beta_j^{(k)} x_{ij} = 0 \quad \forall k \in [b]. \]  
   (2)

3. The symbols satisfy $h$ global constraints given by
   \[ \sum_{i=1}^{m} \sum_{j=1}^{n} \gamma_{ij}^{(k)} x_{ij} = 0 \quad \forall k \in [h]. \]  
   (3)

A setting of $\{\alpha_i^{(k)}\}, \{\beta_j^{(k)}\}, \{\gamma_{ij}^{(k)}\}$ from a field $F \supseteq F_2$ specifies a code $C = C(\{\alpha_i^{(k)}\}, \{\beta_j^{(k)}\}, \{\gamma_{ij}^{(k)}\})$ that instantiates the topology $T_{m \times n}(a, b, h)$.

Intuitively, constraints (1) above ensure that that there are local dependencies in every column; constraints (2) ensure that that there are local dependencies in every row; and constraints (3) serve the goal of providing additional reliability guarantees, if the guarantees provided by (1) and (2) alone are deemed not sufficient. In most settings of interest $a, b$, and $h$ are rather small compared to $m$ and $n$.

**Definition 2.** A failure pattern is a set $E \subseteq [m] \times [n]$ of symbols that are erased. Pattern $E$ is correctable for the topology $T_{m \times n}(a, b, h)$ if there exists a code instantiating the topology where the variables $\{x_{ij}\}_{(i,j) \in E}$ can be recovered from the parity check equations.
Definition 3. A code $C$ that instantiates the topology $T_{m \times n}(a, b, h)$ is Maximally Recoverable (MR) if it corrects every failure pattern that is correctable for the topology.

In other words a code that instantiates a topology is maximally recoverable, if it corrects all erasure patterns that are information theoretically correctable given the topology (dependency) constraints. We now state a basic proposition about such codes, the proof is in Appendix A.

Proposition 4. Let $C$ be an MR instantiation of the topology $T_{m \times n}(a, b, h)$. We have

1. The dimension of $C$ is given by
   \[
   \dim C = (m-a)(n-b) - h; \tag{4}
   \]

2. Let $U \subseteq [m]$, $|U| = m-a$ and $V \subseteq [n]$, $|V| = n-b$ be arbitrary. Then $C|_{U \times V}$ is an $\left[(m-a)(n-b), (m-a)(n-b) - h, h+1\right]$ MDS code. In particular any subset $S \subseteq U \times V$, $|S| = (m-a)(n-b) - h$ is an information set.

3. Assume
   \[
   h \leq (m-a)(n-b) - \max\{(m-a), (n-b)\}; \tag{5}
   \]
   then for all $j \in [n]$, $C$ restricted to column $j$ is an $[m, m-a, a+1]$ MDS code and for all $i \in [m]$, $C$ restricted to row $i$ is an $[n, n-b, b+1]$ MDS code.

The proposition above sheds light on why grid-like topologies arise naturally in distributed storage that are discussed in Section 1. Consider a setting where we have $m$ datacenters, each with $n$ machines where $m \ll n$. One can use a code instantiating $T_{m \times n}(a, b, h)$ to distribute data across the datacenters.

- The code is systematic, item (2) tells us how to select the information symbols. So when no failures happen, the data are readily accessible.

- When up to $a$ datacenters are unavailable, each packet can be recovered by accessing at most $(m-a)$ symbols across the remaining datacenters using the column MDS code. This involves cross datacenter traffic, but since $m$ is small, we do not need to read too many symbols. When fewer than $a$ datacenters are offline, the MDS property implies that any $a$ packets being received suffices for successful decoding.

- When up to $b$ failures happen within a particular datacenter, every packet can be accessed by performing $(n-b)$ reads within the datacenter using the row MDS code.

- The $h$ global parities improve the worst case distance of the code. They are only used when row and column decoding fails, since using them in decoding involves using all the code symbols and hence requires a lot of communication across datacenters.

Grid-like topologies are a unifying framework for several topologies that are used in practice and have been studied in the literature.
1. A maximally recoverable code instantiating a topology closely related to $T_{2 \times 7}(0, 1, 2)$ is used by Microsoft’s Azure storage [HSX+12].

2. A code instantiating $T_{3 \times 14}(1, 4, 0)$ is used by Facebook in its f4 storage system [MLR+14]. The code is the tensor product of a Reed-Solomon code within data centers with a parity check code across data centers.

3. Maximum distance separable (MDS) codes can be viewed as maximally recoverable codes for $T_{1 \times n}(0, 0, h)$. Reed Solomon codes yield explicit constructions over an alphabet of size $n$, and there are lower bounds of $\Omega(n)$ on the field size [Bal12, BB12].

4. The topology $T_{m \times n}(1, 0, h)$ has received a considerable amount of attention, especially in the recent work on LRCs [BHH13, GHJY14, GHSY12, BK15, LL15, HY16]. The work of [GHSY12] showed that this topology is canonical for optimal length codes with given locality and worst-case distance. Correctable patterns for this topology are fully characterized [BHH13, GHJY14]. The best known constructions [GHJY14] are slightly better than random codes in their alphabet size, but are still far from polynomial in the block-length.

3 Our results

We now give formal statements of our three main results.

3.1 A super-polynomial field-size lower bound

Our main result is the first super-polynomial lower bound on the field size required for maximally recoverable codes in any topology. Previously, there was not even a super-linear lower bound known, for any topology. Indeed, our lower bound applies to a very simple topology $T_{n \times n}(1, 1, 1)$. Since $T_{n \times n}(1, 1, 0)$ is just the parity tensor code, $T_{n \times n}(1, 1, 1)$ can be viewed as the parity tensor code with a single global parity check equation added to it.

**Theorem 5.** Any maximally recoverable code for the topology $T_{n \times n}(1, 1, 1)$ requires field size $q = n^{\Omega(\log(n))}$.

The key step in proving this lower bound is the following combinatorial lemma, which might be of independent interest.

**Lemma 6.** Let $\gamma : [n] \times [n] \to \mathbb{F}_2^n$ be a labelling of the edges of the complete bipartite graph $K_{n,n}$ by bit vectors such that for any simple cycle $C$,

$$\sum_{e \in C} \gamma_e \neq 0.$$

Then we have $d \geq \Omega((\log(n))^2)$.

This can be viewed as an instance of the critical problem of Crapo and Rota from the 70s [CR70], where the goal is to find the largest dimensional subspace in $\mathbb{F}_2^N$ that does not intersect a given set $S \subset \mathbb{F}_2^N$, which generalizes the problem of finding the maximum rate binary linear code. Identify $[N]$ with the edges of $K_{n,n}$. Given $\gamma : [N] \to \mathbb{F}_2^n$, the indicators of all sets of edges $E$ such that
\[ \sum_{e \in E} \gamma_e = 0 \] is a subspace, of dimension \( N - d \) or more. Our goal is to find the largest such subspace that does not intersect the set \( S \subseteq \mathbb{F}_2^N \) of indicators of all simple cycles.

Another related problem had been recently studied in [FGT16] in the context of derandomizing parallel algorithms for matching. The authors also consider the problem of assigning weights to edges of a graph, so that simple cycles carry non-zero weight. The key differences form our setting are: we need a single assignment while [FGT16] may have multiple assignments; we care about all simple cycles, while [FGT16] only needed non-zero weights on short cycles; we are interested in fields of characteristic 2 while [FGT16] work in characteristic zero.

### 3.2 Characterizing correctable erasure patterns in tensor products

Perhaps the simplest topology where we cannot characterize correctable patterns of erasures is the topology \( T_{m \times n}(a, b, 0) \), which can be viewed as the tensor of a row and column code, each of which is MDS (according to the last item in Proposition 4). We start with a simple necessary condition for a pattern to be correctable.

**Definition 7.** Consider the topology \( T_{m \times n}(a, b, 0) \) and an erasure pattern \( E \). We say that \( E \) is regular if for all \( U \subseteq [m] \), \( |U| = u \) and \( V \subseteq [n] \), \( |V| = v \) we have
\[
|E \cap (U \times V)| \leq va + ub - ab.
\] (6)

It is not hard to see that regularity is in fact necessary for correctability.

**Lemma 8.** If \( E \) is not a regular pattern, then it is not correctable for \( T_{m \times n}(a, b, 0) \).

**Proof.** Let \( U \times V \) be a sub-grid of \([m] \times [n]\), where (6) is violated. Let \( |U| = u \) and \( |V| = v \). Consider the collection of variables \( \{x_{ij}\} \) from Definition 1. Let us restrict our attention to \( \{x_{ij}\} \) where \( i \in U \) and \( j \in V \) and set all other \( x_{ij} \) to zero. By the last bullet in Section 1.4, the rank of the row and column constraints on the variables \( \{x_{ij}\}_{i \in U, j \in V} \) is at most \( va + ub - ab \).

We next set all \( x_{ij} \)s outside of the set \( E \) to zero. Setting variables outside \( E \) to 0 can only reduce the rank of the row and column constraints further. The number of surviving variables is \( |E| > ua + vb - ab \), since (6) is violated, which exceeds the rank of the constraints. So there exists a codeword supported on \( E \), and \( E \) is not correctable by any code \( C \) that instantiates \( T_{m \times n}(a, b, 0) \).

We conjecture that regularity is also sufficient, and thus yields a characterization of the correctable error patterns in \( T_{m \times n}(a, b, 0) \).

**Conjecture 9.** An erasure pattern \( E \) is correctable for \( T_{m \times n}(a, b, 0) \) if and only if it is regular.

While unable to prove Conjecture 9 in full generality, we establish it for \( a = 1 \). This is a restricted case, but this topology of a row code tensored with a parity check code is one that is of importance in practice [MLR+14].

**Theorem 10.** A pattern \( E \) is correctable for \( T_{m \times n}(1, b, 0) \) if and only if it is regular for \( T_{m \times n}(1, b, 0) \).

Our work on MR codes for \( T_{m \times n}(a, b, 0) \) bears some similarities to the study of weight hierarchies of product codes [SW03, WY93]. The difference is that there one is interested in understanding the relation between weight hierarchies of codes and their tensor products, while we are concerned with minimizing the field size of codes whose tensor products have optimal erasure correction capabilities.
3.3 Asymptotically optimal MR codes for $T_{m \times n}(1, 0, 2)$

The topology $T_{m \times n}(1, 0, h)$ has received a considerable amount of attention, especially in the recent work on LRCs [BHH13, GHJY14, GHSY12, BK15, LL15, HY16]. The work of [GHSY12] showed that this topology is canonical for optimal length codes with given locality and worst-case distance. Correctable patterns for this topology are fully characterized [BHH13, GHJY14]:

**Lemma 11.** A pattern $E \subseteq [m] \times [n]$ is correctable for $T_{m \times n}(1, 0, h)$ if and only if it can be obtained by erasing at most one coordinate in every column of $[m] \times [n]$ and then additionally up to $h$ more coordinates.

For $h = 1$ explicit MR codes exist over a field of size $O(m)$ (which is sub-linear in the input length). For $h \geq 2$ there is a linear $\Omega(mn)$ lower bound on the field size [GHJY14]. Prior to our work this bound was only known to be tight for $h = 2$ and $m = O(1)$, [BHH13, GHJY14]. In the following theorem we remove the assumption $m = O(1)$.

**Theorem 12.** For all $m, n$ there exists an explicit maximally recoverable code instantiating the topology $T_{m \times n}(1, 0, 2)$ over a field of size $O(mn)$.

4 A super-polynomial field-size lower bound

In this section, we prove Theorem 5. The proof is in two steps: we first characterize correctable erasure patterns, and then use this to reduce the lower bound to proving Lemma 6.

4.1 Characterizing correctable erasure patterns in $T_{n \times n}(1, 1, 1)$

Recall that the topology $T_{n \times n}(1, 1, h)$ is defined by the constraints

\[ \forall j \in [n], \sum_{i=1}^{m} \alpha_i x_{ij} = 0, \]  
\[ \forall i \in [m], \sum_{j=1}^{n} \beta_j x_{ij} = 0, \]  
\[ \sum_{i=1}^{m} \sum_{j=1}^{n} \gamma_{ij}^{(k)} x_{ij} = 0 \quad \forall k \in [h]. \]

An assignment of coefficients specifies a code $C = C(\{\alpha_i\}, \{\beta_j\}, \{\gamma_{ij}^{(k)}\})$ that instantiates this topology. We start by showing that the row and column parity equations can be taken to be simple XORs without a loss of generality.

**Lemma 13.** Let $m, n \geq 3$. Assume $h$ is small enough so that (5) holds. There exists an MR instantiation of $T_{m \times n}(1, 1, h)$ where $\alpha_i = 1, \beta_j = 1$ for every $i \in [m], j \in [n]$.

**Proof.** Consider an arbitrary MR instantiation of $T_{m \times n}(1, 1, h)$. By item three in Proposition 4 for every $i, j$, $\alpha_i \neq 0, \beta_j \neq 0$. Let us define new variables $z_{ij} = \alpha_i \beta_j x_{ij}$. Since $\beta_j^{-1} \in \mathbb{F}^*$ is defined,
\[ \beta_j^{-1} z_{ij} = \alpha_i x_{ij}. \] So we can rewrite (7) as

\[ \forall j \in [n], \beta_j^{-1} \cdot \sum_{i=1}^{m} z_{ij} = 0 \]

and hence

\[ \forall j \in [n], \sum_{i=1}^{m} z_{ij} = 0. \] (10)

Similarly, since \( \alpha_i^{-1} \in \mathbb{F}^* \) is defined, we can rewrite (8) as

\[ \sum_{j=1}^{n} z_{ij} = 0. \] (11)

For each \( k \in [h] \), setting \( \gamma_{ij}^{(k)'} = \gamma_{ij}^{(k)} \alpha_i^{-1} \beta_j^{-1} \), we can rewrite (9) as

\[ \sum_{ij} \gamma_{ij}^{(k)'} z_{ij} = 0. \] (12)

It is clear that the code on the \( z_{ij} \)s defined by (10), (11) and (12) can correct the same set of failures as the original code on the \( x_{ij} \)s, so the claim is proved.

For the remainder of this section, we assume that \( h \) is small enough, so that inequality (5) holds and constrain ourselves to considering MR instantiations of \( T_{m \times n}(1, 1, h) \) where \( \alpha_i = 1, \beta_j = 1 \) for every \( i \in [m], j \in [n] \). Such instantiations are specified by setting the coefficients \( \{ \gamma_{ij}^{(k)} \}, k \in [h] \), i.e., we have \( C = C(\{ \gamma_{ij}^{(k)} \}) \).

A failure pattern is given by a subset of edges in the complete bipartite graph \( K_{m,n} \). For each \( (i,j) \in E \) we have variables \( x_{ij} \), which are subject to parity check constraints at each vertex. If a vertex has degree 1, then the parity check lets us solve for the corresponding variable. We iteratively eliminate such vertices, until every remaining vertex in the graph has degree 2 or higher. Let \( E \) denote the set of remanding failures and let \( L, R \) denote the subset of vertices on the two sides that have non-zero degree. Thus we have a bipartite graph \( H(L, R, E) \) where \( \deg(v) \geq 2 \) for every vertex \( v \in L \cup R \). Let \( \ell = |L|, r = |R|, e = |E| \) and let \( c \) denote the number of connected components.

In the topology \( T_{m \times n}(1, 1, h) \), we will refer to (10) and (11) as the XOR constraints, and to (12) as the global constraints.

**Lemma 14.** Using the notation above a failure pattern \( E \) is correctable by \( T_{m \times n}(1, 1, h) \) iff

\[ e \leq h + \ell + r - c. \] (13)

**Proof.** For every edge \( e \in E \), we have a variable \( x_e \). Let \( e \sim v \) denote that \( e \) is incident to \( v \). For every vertex \( v \in L \cup R \) we have the constraint

\[ \sum_{e \sim v} x_e = 0 \] (14)

The number of unknowns is \( e \). To prove the claim, it suffices to show that the rank of the resulting set of constraints is at least \( \ell + r - c + h \). In an MR code, the \( h \) global constraints are
linearly independent of the local XOR constraints at the vertices. So, we will show that the rank of the XOR constraints is $\ell + r - c$. Since each connected component involves disjoint variables on the edges, it suffices to show that the rank of the XOR constraints is $\ell + r - 1$ when there is a single connected component, we can then sum this bound over all the components.

So assume that $H(L, R, E)$ is connected. The upper bound on rank comes from observing that the $\ell + r$ linear constraints satisfy the dependence

$$\sum_{v \in L} \sum_{e \sim v} x_e = \sum_{w \in R} \sum_{e \sim w} x_e$$  \hspace{1cm} (15)$$

since every edge appears exactly once on the LHS and the RHS. We claim that the constraints corresponding to any smaller subset $L' \cup R'$ of vertices are linearly independent. Indeed, we can rewrite a dependence between these constraints as

$$\sum_{v \in L'} \sum_{e \sim v} x_e + \sum_{w \in R'} \sum_{e \sim w} x_e = 0. \hspace{1cm} (16)$$

But since $L' \cup R'$ does not induce a connected subgraph, there must be at least one edge leaving the set, and the corresponding variable appears exactly once. This proves that the rank of the parity check constraints equals $(\ell + r - 1)$.

A simple cycle in $K_{m,n}$ is a connected subgraph where each vertex has degree exactly 2 (in other words, we do not repeat vertices in the cycle).

**Corollary 15.** Correctable failure patterns in $T_{m \times n}(1, 1, 1)$ correspond to simple cycles in $K_{m,n}$. The error pattern corresponding to a simple cycle $C$ is correctable by a particular instantiation $C = C(\{\gamma_e\})_{e \in [m] \times [n]}$ iff

$$\sum_{e \in C} \gamma_e \neq 0. \hspace{1cm} (17)$$

**Proof.** Since every vertex has degree at least 2, we have

$$e \geq \max(2\ell, 2r) \geq \ell + r,$$

with strict inequality whenever some vertex has degree exceeding 2. By Lemma 14,

$$e \leq \ell + r - c + 1 \leq \ell + r$$

with equality iff $c = 1$. Thus if the error pattern $E$ is correctable, it is connected and every vertex in it has degree exactly 2, so $E$ must be a simple cycle.

Let $E$ be a simple cycle. The parity check constraints enforce the condition $x_e = x$ for every $e \in E$. Plugging this into the global parity gives

$$\sum_{e \in E} \gamma_e x = x \sum_{e \in E} \gamma_e = 0.$$

Assuming that $\sum_{e \in E} \gamma_e \neq 0$, the only solution to this system is $x = 0$. This shows that the system of equalities defined by the variables $x_e$ and the parity check equations has a trivial kernel, so it is invertible. \qed
We are now ready to prove Theorem 5 modulo Lemma 6, whose proof appears in the next Section.

**Proof of Theorem 5:** Consider an MR instantiation of $T_{n \times n}(1,1,1)$ over a field $\mathbb{F}_q$ where $q = 2^d$. Assume $n \geq 3$. Now $h = 1$ obviously satisfies (5) and we can apply Lemma 13 to get an MR instantiation over the same field where column and row constraints are simple XORs. Use the global constraint to produce an assignment $\{\gamma_e\}$ of weights to edges of $K_{n,n}$. By Corollary 15 every simple cycle in $K_{n,n}$ now carries a non-zero weight. By Lemma 6 we have $d \geq \Omega((\log n)^2)$. Thus $q \geq n^{\Omega(\log n)}$. □

**4.2 Proof of Lemma 6**

In order to prove Lemma 6, we will consider the more general setting of bounded degree graphs.

We will consider a graph $G = (V,E)$ with maximum degree $D$, where each edge $e \in E$ is assigned a weight $\gamma(e) \in \mathbb{F}$, where $\mathbb{F}$ is a field of characteristic 2. Our result will actually apply to any Abelian group. For path $p$ in the graph, we use $\gamma(p)$ to denote the sum of the weights of all edges in the path. Let $P(v_1,v_2)$ ($v_1 \neq v_2 \in V$) be the set of simple paths from $v_1$ to $v_2$, and $P_k(v_1,v_2)$ be the set of simple paths from $v_1$ to $v_2$ with length $k$, where the length of a path is the number of edges in that path. For a path in $P_k(v_1,v_2)$, we say that $v_1$ is the 1st vertex, $v_2$ is the $(k+1)$-th vertex, and the other $(k-1)$ vertices on the path are the 2nd through the $k$-th vertices according to their positions. We are interested in graphs with the following property:

**Definition 16.** A weighted graph as above satisfies Property $\mathcal{A}$ if for all $v_1 \neq v_2 \in V$ and vertex disjoint simple paths $p_1 \neq p_2 \in P(v_1,v_2)$, their weights satisfy the condition $\gamma(p_1) \neq \gamma(p_2)$.

It is clear that if we assign weights $\gamma$ to the edges of $K_{m,n}$ such that Equation (17) holds, then Property $\mathcal{A}$ holds. We next state the main technical lemma of this section, which shows that there cannot be too many paths with the same length and the same weight.

**Lemma 17.** If $G$ has Property $\mathcal{A}$, then for arbitrary vertices $v_1 \neq v_2 \in V$, positive integer $k \leq \sqrt{D}$, and $\gamma_0 \in \Sigma$, the set

$$S = \{p \in P_k(v_1,v_2) \mid \gamma(p) = \gamma_0\}$$

has cardinality at most $k^{\log_2 k+1}D^{k-\log_2 k-1}$.

**High-level idea:** We think $k$ as a small number for convenience. The goal of the lemma is to show $|S| \lesssim D^{k-\log_2 k-1}$. The total number of paths in $|S|$ would be $D^{k-1}$ if all the intermediate $k-1$ vertices could be chosen ‘freely’. The lemma is saying that we do not have so much ‘freedom’.

We will show that there is a set $T \subseteq S$ with $|T| \gtrsim |S|/k^2$ such that all paths in $T$ share the same $t$-th vertex for some $t \in [2,k]$. That is saying many paths in $S$ are fixed at the $t$-th vertex, and the choice of the $t$-th vertex is not ‘free’. Then we fix the prefix before (or suffix after) the $t$-th vertex and recursively apply the argument to the remaining half of the path. Intuitively, we can do this for $\log_2 k$ rounds (since each time we halve the length of the path) and find $\log_2 k$ vertices that are not ‘free’, which gives the bound $D^{k-\log_2 k-1}$.

The proof is by induction on the length $k$.

**Proof.** Let $f(k) = k^{\log_2 k+1}D^{k-\log_2 k-1}$. We prove by induction on $k$. For $k = 1$, we have $|S| \leq 1 = k^{\log_2 k+1}D^{k-\log_2 k-1} = f(k)$. Assume we have proved the lemma for lengths up to $k-1$, and we consider the case of $k$ $(2 \leq k \leq \sqrt{D})$. 12
If $S = \emptyset$, the lemma is trivial. We only consider the case that $S \neq \emptyset$. We pick an arbitrary path $p_0 \in S$. Then for any other path $p \in S$, $p \neq p_0$, $p$ must intersect $p_0$ at some vertex other than $v_1, v_2$, because of Property $A$. That is, there exists $i, j \in [k - 1]$ such that the $(i + 1)$-th vertex of $p$ is the same as the $(j + 1)$-th vertex of $p_0$. Let $T_{ij}$ denote the set of these paths, formally

$$T_{ij} = \{ p \in S \setminus \{p_0\} \mid \text{the } (i+1)-\text{th vertex of } p \text{ is the } (j+1)-\text{th vertex of } p_0 \}.$$  

Note that $\bigcup_{i,j \in [k-1]} T_{ij} = S \setminus \{p_0\}$. By the Pigeonhole principle, there must exist $i_0, j_0 \in [k - 1]$ such that

$$|T_{i_0,j_0}| \geq \frac{|S| - 1}{(k - 1)^2}.$$

We consider the paths in $T_{i_0,j_0}$. These paths share the same $(i_0 + 1)$-th vertex. We denote this vertex by $v_3$. Every path in $T_{i_0,j_0}$ can be considered as two parts, the head from $v_1$ to $v_3$ (with length $i_0$) and the tail from $v_3$ to $v_2$ (with length $k - i_0$). We will assume that $i_0 \leq k/2$, so that the head not longer than the tail. If this condition does not hold, we can interchange the definition of head and tail.

![Figure 1: Paths in $T_{i_0,j_0}$ are fixed at 3 vertices $v_1, v_2, v_3$.](image)

The number of possible heads equals the number of simple paths from $v_1$ to $v_3$, which is at most $D^{i_0-1}$. We count the paths in $T_{i_0,j_0}$ according to their head. For every choice of head, the weight of the tail is fixed because all paths in $T_{i_0,j_0}$ have the same total weight. Hence by induction hypothesis, the number of possibilities of the tail for every fixed head is bounded by

$$f(k - i_0) = (k - i_0)^{\log_2(k-i_0)+1} D^{k-i_0-\log_2(k-i_0)-1}.$$

So we have

$$|T_{i_0,j_0}| \leq D^{i_0-1} f(k - i_0)
= (k - i_0)^{\log_2(k-i_0)+1} D^{k-\log_2(k-i_0)-2}$$

Hence

$$|S| \leq (k - 1)^2 |T_{i_0,j_0}| + 1
\leq (k - 1)^2 (k - i_0)^{\log_2(k-i_0)+1} D^{k-\log_2(k-i_0)-2} + 1
\leq k^2 (k - i_0)^{\log_2(k-i_0)+1} D^{k-\log_2(k-i_0)-2}.$$  

Let $t = k - i_0$. Since we assume that $i_0 \leq k/2$, we have $k/2 \leq t \leq k$ and

$$|S| \leq k^2 \log_2(t)+1 D^{k-\log_2(t)-2}.$$  

13
We show that the RHS is at most \( f(k) \), by considering the ratio with \( f(k) \),

\[
\frac{k^{2\log_2 t + 1}D^{k - \log_2 k - 2}}{k^{\log_2 k + 1}D^{k - \log_2 k - 1}} = \frac{t^{\log_2 t + 1}}{k^{\log_2 k - 1}D^{\log_2 t - \log_2 k + 1}}
\]

\[
= \left( \frac{kt}{D} \right)^{\log_2 (2t/k)}
\]

\[
= \left( \frac{kt}{D} \right) \log_2 (2t/k)
\]

\[
\leq 1
\]

where in the last step we used the fact \( t \leq k \leq \sqrt{D} \) and \( t \geq k/2 \). Thus \(|S| \leq f(k)\), hence the claim is proved.

We proceed to the proof of the main Lemma.

Proof of Lemma 6: We claim that \( K_{n,n} \) with weights \( \{\gamma(e)\} \) has Property \( \mathcal{A} \). For \( v_1 \neq v_2 \) and vertex disjoint simple paths \( p_1 \neq p_2 \in P(v_1, v_2) \), we can see that \( p_1 \) and \( p_2 \) form a simple cycle. Hence \( \gamma(p_1) + \gamma(p_2) \neq 0 \). Since the alphabet of weights has characteristic 2, we have \( \gamma(p_1) \neq \gamma(p_2) \), and Property \( \mathcal{A} \) is satisfied.

Let \( \ell = \lfloor (\sqrt{n} - 1)/2 \rfloor \) and \( k = 2\ell + 1 \). Clearly, \( k \leq \sqrt{n} \). Pick vertices \( u \) and \( v \) from the two sides of the graph. The number of simple paths from \( u \) to \( v \) is

\[
(n - 1)(n - 1)(n - 2)(n - 2) \cdots (n - \ell)(n - \ell) \geq (n - \ell)^{2\ell}.
\]

Apply Lemma 17 for \( D = n \). Then for every \( \gamma_0 \in \mathbb{F}_2^d \), the number of paths from \( u \) to \( v \) with length \( k \) and weight \( \gamma_0 \) is at most \( n^{\log_2 k + 1}n^{k - \log_2 k - 1} \). Hence we have

\[
2^d \geq \frac{(n - \ell)^{2\ell}}{n^{\log_2 k + 1}n^{k - \log_2 k - 1}}
\]

\[
= \frac{n^{\log_2 k}}{n^{\log_2 k + 1}} \cdot \left( \frac{n - \ell}{n} \right)^{2\ell}
\]

\[
= n^{\log_2 k} \cdot \exp(\Theta(\ell^2/n))
\]

\[
= n^{\Omega(\log n)}
\]

where we used \( k = 2\ell + 1 \) and \( k, \ell = \Theta(\sqrt{n}) \). It follows immediately that \( d = \Omega(\log^2 n) \).

5 Characterizing correctable patterns in \( T_{m \times n}(1, b, 0) \)

In this section, we will prove Theorem 10 in three steps:

1. Lemma 18 shows that it suffices to consider erasure patterns \( E \) where every non-empty row of \( E \) has at least \( b + 1 \) erasures.

2. Lemma 19 establishes the Theorem for erasure patterns \( E \) where every non-empty row has weight exactly \( b + 1 \).
3. Lemma 22 extends the proof to general regular erasure patterns.

Note that by Lemma 8 we only need to argue that regularity is sufficient for MR correctability. To do this, for every regular erasure pattern \( E \subseteq [m] \times [n] \) we need to exhibit a column code \( C_{\text{col}} \) and a row code \( C_{\text{row}} \), so that \( C_{\text{col}} \otimes C_{\text{row}} \) corrects \( E \). Indeed, we can tailor the choice of these codes to the pattern \( E \), and our proof will use the flexibility.

Let \( E \subseteq [m] \times [n] \) be an erasure pattern for \( T_{m \times n}(1, b, 0) \). For \( i \in [m] \), we refer to \( \{(i) \times [n]\} \cap E \) as the \( i \)-th row of \( E \). We often call the number of elements in the \( i \)-th row the weight of the row.

**Lemma 18.** Let \( E \) be an erasure pattern for \( T = T_{m \times n}(1, b, 0) \). Suppose \( E' \subseteq E \) is obtained from \( E \) by restricting \( E \) to rows where \( E \) has \( b+1 \) or more erasures; then \( E \) is correctable for \( T \) iff \( E' \) is correctable for \( T \).

**Proof.** Clearly, if \( E \) is correctable for \( T \) then \( E' \subseteq E \) is also correctable for \( T \). We need to show the converse. So assume that code \( C \) instantiating the topology \( T \) corrects \( E' \). We can assume that \( C \) is maximally recoverable since a maximally recoverable code for this topology will also correct \( E' \). By Proposition 4, each row of \( C \) is an MDS code capable of correcting \( b \) erasures. So we can use row decoding to correct all rows that have \( b \) or fewer erasures, which reduces the problem to correcting \( E' \). By assumption, \( C \) can correct \( E' \), and hence \( E \). \( \square \)

Below is the main technical lemma of this Section.

**Lemma 19.** Let \( E \) be a regular pattern for \( T = T_{m \times n}(1, b, 0) \). Suppose that every row of \([m] \times [n]\) that intersects \( E \), intersects \( E \) in exactly \( b+1 \) locations; then \( E \) is correctable for \( T \).

**Proof.** We fix \( C_{\text{col}} \) to be the simple parity code, i.e., we set all \( \{\alpha_i^{(1)}\}, i \in [m] \) in (1) to one, and focus on constructing the code \( C_{\text{row}} \). Let \( U \times V, |U| = u, |V| = v \) be the smallest enclosing sub-grid for \( E \). In what follows we often find it convenient to represent \( E \) by the bipartite graph \( G \) with node set \( U \cup V \) and edge set representing \( E \) in the natural way. By (6) we have

\[
|E| = u(b+1) \leq v + (u - 1)b.
\]  

(19)

Thus \( u \leq v - b \). Let \( d = (v - b) - u \). We set \( C_{\text{row}} \) to be the linear space spanned by \( (n - v) + (u + d) = n - b \) vectors: \( (n - v) \) unit vectors \( \{e_i\}_{i \in [v]} \backslash V \) and \( u + d \) vectors \( w_1, \ldots, w_u, z_1, \ldots, z_d \in \mathbb{F}^n \) over some large finite field \( \mathbb{F} \). Note that this constitutes a valid choice of the row code as the co-dimension is necessarily at least \( b \). We constrain vectors \( \{w_i\} \) and \( \{z_i\} \) to have no support outside of \( V \). Therefore we often treat these vectors as elements of \( \mathbb{F}^v \) rather than \( \mathbb{F}^n \). Furthermore, for every \( i \in [u] \), we constrain \( w_i \) to have no support outside of the support of the \( i \)-th row of \( E \). Let \( M \in \mathbb{F}^{(u+d) \times v} \) be the matrix whose rows are vectors \( \{w_i\} \) and \( \{z_i\} \). We pick the field \( \mathbb{F} \) to be sufficiently large and select \( \{w_i\} \) and \( \{z_i\} \) at random from \( \mathbb{F}^v \) subject to the support constraints. This allows us to ensure that every minor of \( M \) that can have full rank for some choice of \( \{w_i\} \) and \( \{z_i\} \) indeed has full rank. In particular for all \( U' \subseteq U \) and \( V' \subseteq V \) such that there is a matching of size \( |U'| \) in \( G \) between the nodes \( U' \) and \( V' \) the minor \( M_{U', V'} \) has full rank. Also, all coordinates of \( \{w_i\}_{i \in [u]} \) and \( \{z_i\}_{i \in [d]} \) that do not have to be zero are non-zero.

Below is the key Claim that underlies our proof:

**Claim.** Fix \( j \in [u] \) and consider an arbitrary linear combination \( y \) of vectors \( \{w_i\}_{i \in [u]\backslash j} \) and \( \{z_i\}_{i \in [d]} \) that includes at least one of these vectors with a non-zero coefficient. We claim that

\[
\text{supp}(y) \not\subseteq \text{supp}(w_j).
\]  

(20)
We first prove the claim above and then proceed with the proof of the Lemma. Assume \((20)\) is violated. Let \(U' = U \setminus \{j\}\) and \(V' = V \setminus \text{supp}(w_j)\). It is possible to take a non-trivial linear combination of \(\{w_i\}_{i \in U'}\) and \(\{z_i\}_{i \in [d]}\) that has no support in \(V'\). Observe that \(|U'| + d = u - 1 + d = v - (b + 1) = |V'|\). Therefore existence of a linear combination as above implies that a certain \((u - 1 + d) \times |V'|\) square minor \(M'\) of \(M\) is degenerate.

By discussion preceding \((20)\), we conclude that the restriction of the graph \(G\) to nodes \((U', V')\) has no matching of size \(U'\), as any such matching together with the fact that vectors \(\{z_i\}\) are random with full support could be used to imply that \(M'\) is of full rank. Thus by Hall’s theorem [Juk01, p.55] there exists a set \(U'' \subseteq U'\) such that the size of the neighborhood \(N(U'')\) in \(G\) is at most \(|U''| - \Delta\) for a positive \(\Delta\). Let \(U''' = U'' \cup \{j\}\) and \(V''' = N(U'') \cup \text{supp}(w_j)\). Further, let \(t = |U'''|\). We have
\[
|E \cap (U''' \times V''')| = t \cdot (b + 1).
\] (21)
However
\[
|V'''| + (|U''| - 1) \cdot b = (b + 1 + (t - 1) - \Delta) + (t - 1)b = t(b + 1) - \Delta.
\] (22)
Thus restricting \(E\) to \(U''' \times V'''\) violates \((6)\). This contradiction completes the proof of the Claim. We now use the Claim to prove the Lemma.

Assume for the purpose of contradiction that \(C_{\text{col}} \otimes C_{\text{row}}\) does not correct \(E\). Then \(C_{\text{col}} \otimes C_{\text{row}}\) contains a codeword \(w\) such that \(\text{supp}(w) \subseteq E\). We now make two observations:

- **For all \(i \in U\), the restriction of \(w\) to row \(i\) has be a scalar multiple of the vector \(w_i\) defined above.** This observation follows from the fact that the \(i\)-th row of \(w\) is an element of \(C_{\text{row}}\) and by the Claim above no element of \(C_{\text{row}}\) other than the scalar multiplies of \(w_i\) has its support inside \(\text{supp}(w_i)\).

- **Vectors \(\{w_i\}_{i \in U}\) are linearly independent.** Again this easily follows for the Claim. Every dependency between \(\{w_i\}_{i \in U}\) can be used to obtain a linear combination of \(\{w_i\}_{i \in U \setminus \{j\}}\) whose support falls within \(\text{supp}(w_j)\) for some \(j \in U\).

By the first bullet above, rows of \(w\) are scalar multiples of vectors \(\{w_i\}_{i \in U}\). However, rows of \(w\) are linearly dependent as every column of \(C_{\text{col}} \otimes C_{\text{row}}\) is an element of \(C_{\text{col}}\). We obtain a contradiction with the second bullet above. This completes the proof of the Lemma. \(\Box\)

Lemma 19 suffices to establish Theorem 10 for erasure patterns whose weight is \(b + 1\) across all non-empty rows. We now reduce the general case to this special case. In what follows we often use the same character to denote a row of a topology and the set of erased coordinates of that row. Our reduction is based on the following definition.

**Definition 20.** Let \(E \subseteq [m] \times [n]\) be an erasure pattern for \(T_{m \times n}(1, b, 0)\). Assume that non-empty rows of \(E\) have weights \(b + r_1, \ldots, b + r_u\), where all \(\{r_i\}_{i \in [u]}\) are positive. Set \(\delta = \sum_{i \in [u]} (r_i - 1)\). We define the boosting of \(E\) to be an erasure pattern \(B(E)\) for \(T_{(m+\delta) \times n}(1, b, 0)\) where \(B(E)\) is obtained via the following process:

- **Each row of \(T_{m \times n}(1, b, 0)\) that does not intersect \(E\) yields a row in \(T_{(m+\delta) \times n}(1, b, 0)\) that does not intersect \(B(E)\).**

- **Each row \(s_i\) of \(T_{m \times n}(1, b, 0)\) that intersects \(E\) in \(b + r\) coordinates is replaced by \(r\) rows \(s_{i1}, \ldots, s_{ir}\), where every \(\{s_{ij}\}_{j \in [r]}\) contains the first \(b\) elements of \(s_i\); the weight of each \(\{s_{ij}\}_{j \in [r]}\) is \(b + 1\); and the union of supports of all \(\{s_{ij}\}_{j \in [r]}\) is the support of \(s_i\).**
We demonstrate the concept of boosting by the following example.

**Example 21.** A pattern \( E \) for \( T_{2 \times 5}(1, 2, 0) \) and the boosted pattern \( B(E) \) for \( T_{4 \times 5}(1, 2, 0) \).

\[
E = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \Rightarrow B(E) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}
\] (23)

The following lemma shows that boosting preserves regularity.

**Lemma 22.** Let \( E \subseteq [m] \times [n] \) be an erasure pattern for \( T = T_{m \times n}(1, b, 0) \) where every non-empty row has weight \( b + 1 \) or more. Let \( E' = B(E) \) be the boosting of \( E \) viewed as an erasure pattern for \( T' = T_{m' \times n}(1, b, 0) \). If \( E \) is regular; then \( E' \) is also regular.

**Proof.** Let \( U' \times V \) be an arbitrary sub-grid of \( T' \). Let \( |U'| = u', |V| = v \). Note that rows of \( T' \) that arise by boosting the rows of \( E \) have two indices \((i, j)\). Let \( U = \{ i \in [m] \mid \exists j : (i, j) \in U' \} \). Let \( |U| = u \). Note that \( U \times V \) is a sub-grid of \( T \). In what follows we argue that \( E' \) does not violate (6) on \( U' \times V \); since \( E \) does not violate (6) on \( U \times V \). Consider

\[
\Delta(E') = |E' \cap (U' \times V)| - (v + (u' - 1)b) = \sum_{(i,j) \in U'} (|s_{ij} \cap V| - b) - (v - b) = \sum_i \sum_{j: (i,j) \in U'} (|s_{ij} \cap V| - b) - (v - b).
\] (24)

We claim that for all \( i \in [m] \):

\[
\sum_{j: (i,j) \in U'} (|s_{ij} \cap V| - b) \leq |s_i \cap V| - b.
\] (25)

To see this assume that \( V \) intersects the sets of \( b \) first elements of \( w_i \) in \( c \leq b \) coordinates. Let the sum above include \( t \) terms. Expression on the left simplifies to \( tc + t' - tb \), for \( t' \leq t \). Expression on the right simplifies to \( c + t' - b \). It remains to note that

\[ tc + t' - tb \leq c + t' - b \quad \text{as} \quad c \leq b. \]

and (25) follows. Combining (24) and (25) we conclude that

\[
\Delta(E') \geq \sum_i (|s_i \cap V| - b) - (v - b) = \Delta(E) \geq 0.
\]

This completes the proof of the Lemma.

**Proof of Theorem 10:** By Lemma 18 we can assume that every row of \( E \) has weight at least \( b + 1 \). Consider the boosted pattern \( B(E) \) for \( T' = T_{m' \times n}(1, b, 0) \). By Lemma 22, \( B(E) \) is regular. Thus by Lemma 19, \( B(E) \) is correctable for \( T' \). Let \( C_{\text{col}} \otimes C_{\text{row}} \) be the instantiation that corrects \( B(E) \) obtained via Lemma 19. Note that \( C_{\text{col}} \) is the simple parity check code that we denote by \( P_{m'} \).

We claim that the tensor product of the parity check code \( P_m \) with \( C_{\text{row}} \) corrects \( E \) for \( T \). Assume the contrary. Let \( w \) be the codeword of \( P_m \otimes C_{\text{row}} \) where \( \text{supp}(w) \subseteq E \). Let \( u \) be the
number of non-zero rows in $E$. For $i \in [u]$, let $s_i$ be the $i$-th non-zero row of $w$. Assume for all $i \in [u]$, the weight of $s_i$ is $b + r_i$. We now use $w$ to obtain a codeword $w'$ that resides on $B(E)$ for $P_{m'} \otimes C_{\text{row}}$ instantiation of $T'$. Our construction is based on the following observation:

$$\dim \left( C_{\text{row}} \big| \text{supp}(w_i) \right) = r_i. \quad (26)$$

We now prove the observation. Firstly, consider vectors $\{w_{ij}\}_{j \in [r_i]}$ with supports $\{s_{ij}\}_{j \in [r_i]}$ that are used in the construction of the linear space $C_{\text{row}}$ in the proof of Lemma 19. These $r_i$ vectors are clearly linearly independent given their support structure. Secondly, note that if $C_{\text{row}}$ had any vector, other then the linear combinations of $\{w_{ij}\}_{j \in [r_i]}$, reside on $\text{supp}(w_i)$; then we would immediately get a contradiction to the key Claim (20) in the proof of Lemma 19.

Using the observation and the argument above we can represent every $w$ as a unique linear combination of vectors $\{w_{ij}\}_{j \in [r_i]}$ with supports $\{s_{ij}\}_{j \in [r_i]}$. Now the collection of vectors $\{w_{ij}\}_{i \in [u], j \in [r_i]}$ yields a codeword $w'$ of $P_{m'} \otimes C_{\text{row}}$ that resides entirely on $B(E)$, contradicting the fact that $B(E)$ is correctable for $T'$.

\[\square\]

## 6 Maximally recoverable codes over linear fields for $T_{m\times n}(1, 0, 2)$

In this section, we prove Theorem 12.

**Proof of Theorem 12:** Let $M$ be the smallest power of 2 that is no less than $m$ and $N$ be the smallest power of 2 that is no less than $n$. We now present an explicit MR instantiation of $T_{m\times n}(1, 0, 2)$ over $F_{MN}$:

- We set all $\{\alpha_i^{(1)}\}_{i \in [m]}$ in (1) to be equal to one.

- To complete the specification of the code we need to specify $\{\gamma_{ij}^{(1)}, \gamma_{ij}^{(2)}\}_{i \in [m], j \in [n]} \in F_{MN}$. In order to do so let us fix a set $\{s_1, s_2, \ldots, s_n\} \subseteq F_{MN}$ to be a subset of an additive subgroup $G \subseteq F_{MN}$ of size $M$ and $c_1, \ldots, c_n \in F_{MN}$ to be field elements, such that $c_{j_1} \not\equiv c_{j_2} + G$, for $j_1 \neq j_2$. (In other words $\{c_j\}_{j \in [n]}$ belong to different cosets of $F_{MN}$ modulo the subgroup $G$.)

For $i \in [m], j \in [n]$ we set

$$\gamma_{ij}^{(1)} = s_i, \quad \gamma_{ij}^{(2)} = s_i^2 + c_j \cdot s_i. \quad (27)$$

By Lemma 11 it suffices to show that every pattern of erasures obtained by erasing one location per column and two more arbitrary locations is correctable by our instantiation of $T_{m\times n}(1, 0, 2)$. Note that every column that carries just one erasure easily corrects this erasure since all $\alpha_i^{(1)} = 1$.

We consider two cases:

- **Some column $j \in [n]$ carries three erasures.** Assume erasures are in rows $i_1, i_2$ and $i_3$. Solving linear system (1), (3) amounts to inverting a $3 \times 3$ matrix, whose determinant is non-zero:

$$\det \begin{pmatrix} 1 & 1 & 1 \\ s_{i_1} & s_{i_2} & s_{i_3} \\ s_{i_1}^2 + c_{j_1} \cdot s_{i_1} & s_{j_2}^2 + c_{j_2} \cdot s_{j_2} & s_{i_3}^2 + c_{j_3} \cdot s_{i_3} \end{pmatrix} = \det \begin{pmatrix} 1 & 1 & 1 \\ s_{i_1} & s_{i_2} & s_{i_3} \\ s_{i_1}^2 & s_{i_2}^2 & s_{i_3}^2 \end{pmatrix} \neq 0.$$

Therefore the erasure pattern is correctable.
There are two distinct columns $j_1, j_2 \in [n]$ each carrying two erasures. Assume column $j_1$ has erasures in rows $i_1$ and $i_2$, while column $j_2$ has erasures in rows $i_3$ and $i_4$. This time solving linear system (1), (3) amounts to inverting a $4 \times 4$ matrix, whose determinant is again non-zero:

$$\det\begin{pmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
s_{i_1}^2 + c_{j_1} \cdot s_{i_1} & s_{i_2}^2 + c_{j_1} \cdot s_{i_2} & s_{i_3}^2 + c_{j_1} \cdot s_{i_3} & s_{i_4}^2 + c_{j_1} \cdot s_{i_4} \\
s_{i_1}^2 + c_{j_2} \cdot s_{i_1} & s_{i_2}^2 + c_{j_2} \cdot s_{i_2} & s_{i_3}^2 + c_{j_2} \cdot s_{i_3} & s_{i_4}^2 + c_{j_2} \cdot s_{i_4}
\end{pmatrix}$$

$$= \det\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
s_{i_1}^2 + s_{i_2} & s_{i_1}^2 + s_{i_2} & s_{i_3}^2 + s_{i_4} & s_{i_3}^2 + s_{i_4} \\
s_{i_1}^2 + c_{j_1} \cdot s_{i_1} & (s_{i_1} + s_{i_2})^2 + c_{j_1} \cdot (s_{i_1} + s_{i_2}) & s_{i_3}^2 + c_{j_2} \cdot s_{i_3} & (s_{i_3} + s_{i_4})^2 + c_{j_2} \cdot (s_{i_3} + s_{i_4})
\end{pmatrix}$$

$$= \det\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
s_{i_1}^2 + s_{i_2} & s_{i_1}^2 + s_{i_2} & s_{i_3}^2 + s_{i_4} & s_{i_3}^2 + s_{i_4} \\
s_{i_1}^2 + c_{j_1} \cdot s_{i_1} & (s_{i_1} + s_{i_2})^2 + c_{j_1} \cdot (s_{i_1} + s_{i_2}) & s_{i_3}^2 + c_{j_2} \cdot s_{i_3} & (s_{i_3} + s_{i_4})^2 + c_{j_2} \cdot (s_{i_3} + s_{i_4})
\end{pmatrix}$$

$$= (s_{i_1} + s_{i_2}) (s_{i_3} + s_{i_4}) (s_{i_1} + s_{i_2} + s_{i_3} + s_{i_4} + c_{j_1} + c_{j_2} + 1) 
eq 0.$$ 

In the last step, we used the fact that $s_{i_1} + s_{i_2} + s_{i_3} + s_{i_4} + c_{j_1} + c_{j_2} = 0$. This follows from $s_{i_1} + s_{i_2} + s_{i_3} + s_{i_4} \in G$ and $c_{j_2} \not\in c_{j_1} + G$.

\[\square\]

## 7 Open problems

The theory of maximally recoverable codes is in its infancy. There is a wide array of questions that remain open. Here we highlight some of the prominent ones:

1. The topology $T_{m \times n}(1, 0, h)$ is well studied in the literature and used in practice. Yet the alphabet size of MR codes for this topology is poorly understood. There is a linear $\Omega(mn)$ lower bound that applies when $h \geq 2$. For $h = 2$ this bound is asymptotically tight by Theorem 12. For $h \geq 3$ the best constructions [GHJY14] use alphabet of size $O((mn)^c)$ for constants $c < 1$. Obtaining a super-linear lower bound or improving the upper bound would be of great interest.

2. There does not seem to be a way to generalize the above construction from [GHJY14] beyond $T_{m \times n}(a, 0, h)$. Obtaining explicit MR codes for other topologies such as $T_{m \times n}(1, 1, 1)$ or $T_{m \times n}(1, b, 0)$ is an intriguing open problem.

3. Can one generalize our Theorem 5 to get a lower bound of $\exp(\log m \cdot \log n)$ for $T_{m \times n}(1, 1, 1)$? We do not know how to get a better bound than $\exp((\min(\log m, \log n))^2)$. More ambitiously, can one obtain a lower bound of $\exp(n^{O(1)})$ for some $T_{m \times n}(a, b, h)$?

4. Establish Conjecture 9 regarding correctable error patterns in $T_{m \times n}(a, b, h)$ for $a > 1$. What are the correctable erasure patterns for $T_{m \times n}(a, b, h)$ for general $h > 0$?
Acknowledgements

We express our deep gratitude to Swastik Kopparty who was a part of the project in the early stages of the work but chose not to be included on the paper.

References

[Bal12] Simeon Ball. On sets of vectors of a finite vector space in which every subset of basis size is a basis. *Journal of European Mathematical Society*, 14:733–748, 2012. 3

[BB12] Simeon Ball and Jan De Beule. On sets of vectors of a finite vector space in which every subset of basis size is a basis (II). *Designs Codes and Cryptography*, 65(1-2):5–14, 2012. 3

[BHH13] Mario Blaum, James Lee Hafner, and Steven Hetzler. Partial-MDS codes and their application to RAID type of architectures. *IEEE Transactions on Information Theory*, 59(7):4510–4519, 2013. (document), 1.1, 1.2, 4, 3.3, 3.3

[BK15] S. B. Balaji and P. Vijay Kumar. On partial maximally-recoverable and maximally-recoverable codes. arXiv:1501.07130, 2015. 4, 3.3

[Bla13] Mario Blaum. Construction of PMDS and SD codes extending RAID 5. Arxiv 1305.0032, 2013. 1.2

[CHL07] Minghua Chen, Cheng Huang, and Jin Li. On maximally recoverable property for multi-protection group codes. In *IEEE International Symposium on Information Theory (ISIT)*, pages 486–490, 2007. 2

[CR70] H. Crapo and G.C. Rota. *Combinatorial Geometries*. M.I.T. Press, Cambridge, 1970. 3.1

[DGW+10] Alexandros G. Dimakis, Brighten Godfrey, Yunnan Wu, Martin J. Wainwright, and Kannan Ramchandran. Network coding for distributed storage systems. *IEEE Transactions on Information Theory*, 56(9):4539–4551, 2010. 1

[FGT16] Stephen Fenner, Rohit Gurjar, and Thomas Thierauf. Bipartite perfect matching is in quasi-NC. In *48th ACM Symposium on Theory of Computing (STOC)*, 2016. 3.1

[GHJY14] Parikshit Gopalan, Cheng Huang, Bob Jenkins, and Sergey Yekhanin. Explicit maximally recoverable codes with locality. *IEEE Transactions on Information Theory*, 60(9):5245–5256, 2014. (document), 2, 1.1, 1.2, 4, 3.3, 3.3, 1, 2

[GHSY12] Parikshit Gopalan, Cheng Huang, Huseyin Simitci, and Sergey Yekhanin. On the locality of codeword symbols. *IEEE Transactions on Information Theory*, 58(11):6925–6934, 2012. 1, 1.2, 4, 3.3

[HCL07] Cheng Huang, Minghua Chen, and Jin Li. Pyramid codes: flexible schemes to trade space for access efficiency in reliable data storage systems. In *6th IEEE International Symposium on Network Computing and Applications (NCA 2007)*, pages 79–86, 2007. 1.2
We proceed item by item.

1. We first argue that \( \dim C \geq (m-a)(n-b) - h \). To see this note that constraints from groups one and two in Definition 1 yield a tensor product of linear codes \( C_{\text{col}} \) and \( C_{\text{row}} \), where \( \dim C_{\text{col}} \geq m-a \) and \( \dim C_{\text{row}} \geq n-b \). We have,

\[
\dim C_{\text{col}} \otimes C_{\text{row}} = \dim C_{\text{col}} \cdot \dim C_{\text{row}} \geq (m-a)(n-b). \tag{28}
\]
Adding group 3 constrains can reduce the dimension by no more than $h$.

Now assume $\dim C > (m-a)(n-b) - h$. Consider another instantiation $C'$ of $T_{m \times n}(a,b,h)$ where $\dim C'_\text{col} = m - a$, $\dim C'_\text{row} = n - b$, and thus constraints from groups one and two yield a code of dimension $(m-a)(n-b) - h$. Let $S \subseteq [m] \times [n]$ be the information set for that code. Set constrains in group 3 to be linearly independent and have no support outside of $S$. This implies $\dim C' = (m-a)(n-b) - h$. Let $E \subseteq [m] \times [n]$ be a complement of the information set of $C'$. It is easy to see that $C'$ recovers $E$. However $C$ cannot recover $E$ as $\dim C > mn - |E|$.

2. Fix $C'_\text{col}$ and $C'_\text{row}$ to be MDS codes. Clearly, $U$ is an information set of $C'_\text{col}$ and $V$ is an information set of $C'_\text{row}$. Therefore $U \times V$ is an information set of $C'_\text{col} \otimes C'_\text{row}$. To complete the specification of $C'$, fix constrains in group 3 to have no support outside of $U \times V$ and define an MDS code of co-dimension $h$ on $U \times V$. Now $C'|_{U \times V}$ is an MDS code of dimension $(m-a)(n-b) - h$. Thus $C'|_{U \times V}$ also has to be an MDS code of dimension $(m-a)(n-b) - h$, as $C$ corrects every pattern of erasures that is corrected by $C'$.

3. Our goal here is to show that under the mild technical assumption (5) both the row code and the column code have to be MDS\(^1\). If suffices to prove this claim for column codes. Suppose $C$ is an MR instantiation but the column code $C_{\text{col}}$ is not MDS. This implies that there is some subset $U \subseteq [m]$, $|U| = m - a$ that is not an information set. In particular, there exists a linear dependence between the symbols of $C_{\text{col}}|U$. Let $V \subseteq [n]$, $|V| = n - b$ be arbitrary. By the item above, for all $h$-sized sets $H \subseteq U \times V$, the set $(U \times V) \setminus H$ is an information set for $C$. By (5) it is possible to pick $H$ so that $(U \times V) \setminus H$ contains a complete column of $U \times V$. In such case $(U \times V) \setminus H$ cannot be an information set as entries of the column are linearly dependent, and we arrive at a contradiction.

\[^1\]One can show that in general the converse is not true; a tensor product of two MDS codes is not necessarily maximally recoverable.