# Computation Efficient Coded Linear Transform: Supplement

**Sinong Wang**[1], **Jiashang Liu**[1], **Ness Shroff**[1,2] **Pengyu Yang**[3]
[1]Department of Electrical and Computer Engineering, The Ohio State University
[2]Department of Computer Science and Engineering, The Ohio State University
[3]Department of Mathematics, The Ohio State University

## A Proof of Theorem 1

Given parameters $m$ and $n$, considering any coded computation scheme that can resist $s$ stragglers. We first define the following bipartite graph model between $m$ workers indexed by $[m]$ and $n$ data partitions indexed by $[n]$, where we connect node $i \in [m]$ and node $j \in [n]$ if $m_{ij} \neq 0$ (worker $i$ has access to data block $\mathbf{A}_j$). The degree of worker node $i \in [m]$ is $\|\mathbf{M}_i\|_0$. We next show by contradiction that the degree of node $j \in [n]$ must be at least $s+1$. Suppose that it is less than $s+1$ and all its neighbors are stragglers. In this case, there exists no worker that is a non-straggler and also access to $\mathbf{A}_j$(or the corresponding submatrix of coding matrix is rank deficient). Hence, it contradicts the assumption that it can resist $s$ stragglers.

Based on the above argument and the fact that, the sum of the degrees of one partition is equal to the sum of degrees in another partition in the bipartite graph, we have that the computation load

$$l(\mathbf{M}) = \sum_{i=1}^{n} \|\mathbf{M}_i\|_0 \geq n(s+1). \tag{17}$$

Therefore, the theorem follows.

## B Proof of Theorem 3

Based on our construction, the cardinality of the set

$$|[n] \backslash \{i_1, \ldots, i_k\}| = n - k. \tag{18}$$

Since $n < i_{k+1} < \cdots < i_n \leq n+s$, we have $n-k \leq s$. We next show, after we recover the blocks indexed by $[n]\backslash\{i_1, \ldots, i_k\}$, the rest blocks can be recovered by peeling decoding without rooting steps. Combining these results together, the total number of rooting steps is at most $s$.

Since we utilize the rooting step to recover blocks indexed by $[n]\backslash\{i_1, \ldots, i_k\}$, we obtain that matrix $\mathbf{M}$ $i$th column $\mathbf{M}_i = \mathbf{0}$ for $i \in \{1, \ldots, i_1 - 1\}$. Based on our construction of the $s$-diagonal code, we have $m_{i_1 i_1} \neq 0$, which implies $i_1$th block is a ripple. Then we can use the result $\tilde{\mathbf{y}}_{\mathbf{i_1}}$ to recover block $\mathbf{y}_{\mathbf{i_1}}$ and peel the $i_1$th column, which implies that $\mathbf{M}_{i_1} = \mathbf{0}$. Using the similar process, we can find a ripple $\tilde{\mathbf{y}}_{\mathbf{i_2}}$ and peel the $i_2$th column. Continue this process, we can peel the $i_k$th column.

Here we analyze the complexity of above procedure. During each iteration, the complexity of operation $\tilde{\mathbf{y}}_j = \tilde{\mathbf{y}}_j - m_{ji}\mathbf{A}_i\mathbf{x}$ is $O(r/n)$. There exists the total $n(s+1)$ the above operations. The complexity from peeling decoding is $r(s+1)$. The complexity in $s$ rooting steps (16) is $O(rs)$. Therefore, the total complexity is $O(rs)$ and theorem follows.

## C Proof of Lemma 3

A direct application of Hall's theorem is: given a bipartite graph $G^D(V_1, V_2)$, for each $U \subseteq [m]$ with $|U| = n$, each subgraph $G^D(U, V_2)$ contains a prefect matching if and only if every subset set $S \subseteq V_1$ such that $|N(S)| < |S|$, where the neighboring set $N(S)$ is defined as $N(S) = \{y | x, y \text{ are connected for some } x \in S\}$. This result is equivalent to the following condition: for each subset $I \subseteq [m]$,

$$\left| \bigcup_{i \in I} \text{supp}(\mathbf{M}_i) \right| \geq |I|, \tag{19}$$

where $\text{supp}(\mathbf{M_i})$ is defined as the support set: $\text{supp}(\mathbf{M_i}) = \{j | m_{ij} \neq 0, j \in [n]\}$, $\mathbf{M_i}$ is $i$th row of the coding matrix $\mathbf{M}$. Suppose that the set $I = \{i_1, i_2, \ldots, i_k\}$ with $i_1 < i_2 < \ldots < i_k$ and $\text{supp}(\mathbf{M}_{i_{k_1}}) \cap \text{supp}(\mathbf{M}_{i_{k_2}}) \neq \emptyset$. Otherwise, we can divide the set into two parts $I_l = \{i_1, i_2, \ldots, i_{k_1}\}$ and $I_r = \{i_{k_2}, i_2, \ldots, i_k\}$ and prove a similar result in these two sets. Based on our construction of diagonal code, we have

$$\left| \bigcup_{i \in I} \text{supp}(\mathbf{M}_i) \right| = \min\{i_k, n\} - \max\{1, i_1 - s\} \overset{(a)}{\geq} k, \tag{20}$$

The above, step (a), is based on the fact that $i_k - i_1 \geq k$. Therefore, the lemma follows.

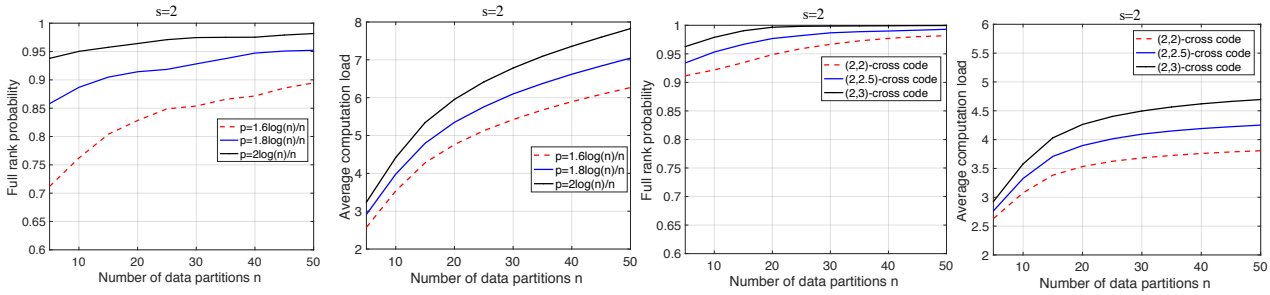


Figure 5: Statistical convergence speed of full rank probability and average computation load of random code under number of stragglers $s = 2$.

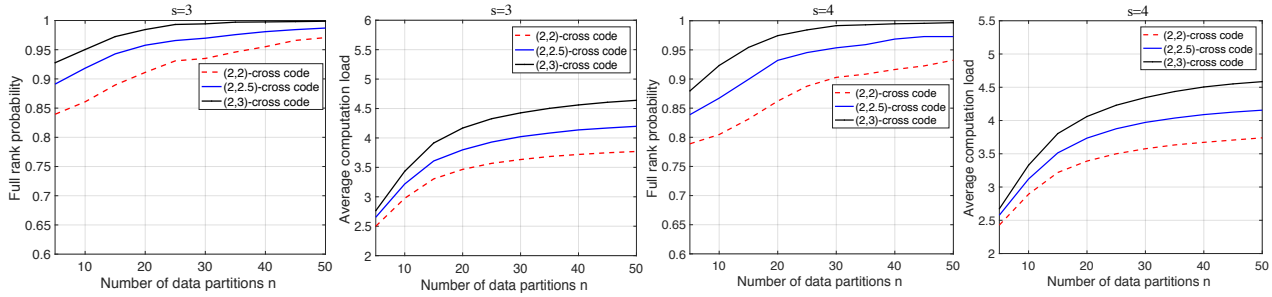


Figure 6: Statistical convergence speed of full rank probability and average computation load of $(d_1, d_2)$-cross code under number of stragglers $s = 3, 4$.

## D Proof of Corollary 1

To prove that the 1-diagonal code achieves the recovery threshold $n$, we need to show that, for each subset $U \subseteq [n+1]$ with $|U| = n$, submatrix $\mathbf{M}^U$ is full rank. Let $U = [n+1]\backslash\{k\}$, the submatrix $\mathbf{M}^U$ satisfies

$$\mathbf{M}^U = \begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}, \end{bmatrix} \tag{21}$$

where $\mathbf{E}$ is a $(k-1)$ dimensional square submatrix consisting of first $(k-1)$ rows and columns, and $\mathbf{F}$ is a $(n-k+1)$ dimensional square submatrix consists of the last $(n-k+1)$ rows and columns. The matrix $\mathbf{E}$ is a lower diagonal matrix due to the fact that, for $i < j$,

$$E_{ij} = m_{ij} = 0. \tag{22}$$

The matrix $\mathbf{E}$ is a upper diagonal matrix due to the fact that, for $i > j$,

$$F_{ij} = m_{i+k,j+k-1} \overset{(a)}{=} 0. \tag{23}$$

The above, (a) utilizes the fact that $(i+k)-(j+k-1) \geq 2$ when $i > j$. Based on the above analysis, we have

$$\det(\mathbf{M}^U) = \det(\mathbf{E}) \cdot \det(\mathbf{F}) = 1, \tag{24}$$

which implies that matrix $\mathbf{M}^U$ is full rank. Therefore, the corollary follows.

## E Numerical Results of Random Code

We examine the performance of the proposed $p$-Bernoulli code and $(d_1, d_2)$-cross code in terms of the convergence speed of full rank probability and computation load. In Fig. 5 and Fig. 5 , we plot the percentage of the full rank $n \times n$ square submatrix and the average computation load $l(\mathbf{M})/m$ of each scheme, based on 1000 experimental runs. Each column of the $(2, 2.5)$-code independently and randomly chooses 2 or 3 nonzero elements with equal probability. It can be observed that the full rank probability of both $p$-Bernoulli code and $(d_1, d_2)$-cross code converges to 1 for relatively small values of $n$. The $(d_1, d_2)$-cross code exhibits even faster convergence and much less computation load compared to the $p$-Bernoulli code. For example, when $n = 20, s = 4$, the $(2, 2)$-cross code achieves the full rank probability of 0.86 and average computation load 3.4. This provides evidence that $(d_1, d_2)$-cross code is useful in practice. Moreover, in practice, one can use random codes by running multiple rounds of trails to find a "best" coding matrix with even higher full rank probability and lower computation load.

## F Proof of Theorem 5

Based on our analysis in Section 4, the full rank probability of an $n \times n$ submatrix $\mathbf{M}^U$ can be lower bounded by a constant times the probability of the existence of a perfect matching in a bipartite graph.

$$\mathbb{P}(|\mathbf{M}^U| \neq 0) = \underbrace{\mathbb{P}(|\mathbf{M}^U| \neq 0 \big| |\mathbf{M}^U(x)| \not\equiv 0)}_{\text{S-Z Lemma: } \geq 1-1/2C_m^n} \cdot \underbrace{\mathbb{P}(|\mathbf{M}^U(x)| \not\equiv 0)}_{\text{contains perfect matching}} +$$

$$\underbrace{\mathbb{P}(|\mathbf{M}^U| \neq 0 \big| |\mathbf{M}^U(x)| \equiv 0)}_{0} \cdot \mathbb{P}(|\mathbf{M}^U(x)| \equiv 0) \quad (25)$$

Therefore, to prove that the $p$-Bernoulli code achieves the probabilistic recovery threshold of $n$, we need to show that each subgraph contains a perfect matching with high probability. Without loss of generality, we can define the following random bipartite graph model.

**Definition 10.** ($p$-Bernoulli random graph) *Graph $G^b(U, V_2, p)$ initially contains isolated nodes with $|U| = |V_2| = n$. Then each node $v_1 \in U$ and node $v_2 \in V_2$ is connected with probability $p$ independently.*

Clearly, the above model describes the support structure of each submatrix $\mathbf{M}^U$ of $p$-Bernoulli code. The rest is to show that, with specific choice of $p$, the subgraph $G^b(U, V_2, p)$ contains a perfect matching with high probability.

The technical idea is to use Hall's theorem. Assume that the bipartite graph $G^b(U, V_2, p)$ does not have a perfect matching. Then by Hall's condition, there exists a violating set $S \subseteq U$ or $S \subseteq V_2$ such that $|N(S)| < |S|$. Formally, by choosing such an $S$ having smallest cardinality, one immediate consequence is the following technical statement.

**Lemma 5.** *If the bipartite graph $G^b(U, V_2, p)$ does not contain a perfect matching, then there exists a set $S \subseteq U$ or $S \subseteq V_2$ with the following properties.*

1. $|S| = |N(S)| + 1$.
2. *For each node $t \in N(S)$, there exists at least two adjacent nodes in $S$.*
3. $|S| \leq n/2$.

**Case 1:** We consider $S \subseteq U$ and $|S| = 1$. In this case, we have $|N(S)| = 0$ and need to estimate the probability that there exists one isolated node in partition $U$. Let random variable $X_i$ be the indicator function of the event that node $v_i$ is isolated. Then we have the probability that

$$\mathbb{P}(X_i = 1) = (1-p)^n,$$

Let $X$ be the total number of isolated nodes in partition $U$. Then we have

$$\mathbb{E}[X] = E\left[\sum_{i=1}^{n} X_i\right] = n(1-p)^n \overset{(a)}{\leq} \frac{1}{n}. \quad (26)$$

The above, step (a) utilizes the assumption that $p = 2\log(n)/n$ and the inequality that $(1 + x/n)^n \leq e^x$.

**Case 2:** We consider $S \subseteq U$ and $2 \leq |S| \leq n/2$. Let $E$ be the event that such an $S$ exists, we have

$$\begin{aligned}
\mathbb{P}(\mathbb{E}) \leq & \sum_{k=2}^{n/2} \binom{n}{k} \binom{n}{k-1} \binom{k}{2}^{k-1} (1-p)^{k(n-k+1)} p^{2(k-1)} \\
\overset{(a)}{<} & \sum_{k=2}^{n/2} \frac{1}{6} \cdot \frac{n}{(n-k)(n-k+1)} \cdot \frac{n^{2n}}{k^{2k}(n-k)^{2(n-k)}} \\
& \left[\frac{k(k-1)}{2}\right]^{k-1} \\
& (1-p)^{k(n-k+1)} p^{2(k-1)} \\
\overset{(b)}{<} & \sum_{k=2}^{n/2} \frac{e^2 n}{6k^2(n-k)(n-k+1)} \cdot \left(\frac{2\log^2(n)}{n}\right)^{k-1} \\
\overset{(c)}{<} & \sum_{k=2}^{n/2} \frac{2}{3(n-1)} \cdot \left(\frac{2\log^2(n)}{n}\right)^{k-1} \\
< & \frac{\log^2(n)}{3n}. \quad (27)
\end{aligned}$$

The above, step (a) is based on the inequality

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n \leq n! \leq \frac{60}{59}\sqrt{2\pi n}\left(\frac{n}{e}\right)^n, \forall n \geq 5. \quad (28)$$

The step (b) utilizes the fact that $p = 2\log(n)/n$, $k \leq n/2$ and the inequality $(1 + x/n)^n \leq e^x$; step (c) is based on the fact that $k(n-k+1) \geq 2(n-1)$ and $k(n-k) \geq 2(n-2)$, $n/(n-2) < 5/3$ for $k \geq 2$ and $n \geq 5$. Utilizing the union bound to sum the results in case 1 and case 2, we can obtain that the probability that graph $G(U, V_2, p)$ contains a perfect matching is at least

$$1 - \frac{\log^2(n)}{3n}. \quad (29)$$

Therefore, incorporating this result into estimating (25), the theorem follows.

## G Proof of Theorem 6

To prove the $(d_1, d_2)$-cross code achieves the probabilistic recovery threshold of $n$, we need to show that each subgraph of the following random bipartite graph contains a perfect matching with high probability.

**Definition 11.** ($(d_1, d_2)$-regular random graph) *Graph $G^c(V_1, V_2, d_1, d_2)$ initially contains the isolated nodes with $|V_1| = m$ and $|V_2| = n$. Each node $v_1 \in V_1$ ($v_2 \in V_2$) randomly and uniformly connects to $d_1$ ($d_2$) nodes in $V_1$ ($V_2$).*

The corresponding subgraph is defined as follows.

**Definition 12.** *For each $U \subseteq V_1$ with $|U| = n$, the subgraph $G^c(U, V_2, d_1, \bar{d})$ is obtained by deleting the nodes in $V_1 \backslash U$ and corresponding arcs.*

Clearly, the above definitions of $(d_1, d_2)$-regular graph and corresponding subgraph describe the support structure of the coding matrix $\mathbf{M}$ and submatrix $\mathbf{M}^U$ of the $(d_1, d_2)$-cross code. Moreover, we have the following result regarding the structure of each subgraph $G^c(U, V_2, d_1, \bar{d})$

**Claim.** For each $U \subseteq V_1$ with $|U| = n$, the subgraph $G^c(U, V_2, d_1, \bar{d})$ can be constructed from the following procedure" (i) Initially, graph $G^c(U, V_2, d_1, \bar{d})$ contain the isolated nodes with $|U| = |V_2| = n$; (ii) Each node $v_1 \in U$ randomly and uniformly connects to $d_1$ nodes in $V_2$; (iii) Each node $v_2 \in V_2$ randomly and uniformly connects to $l$ nodes in $V_1$, where $l$ is chosen according to the distribution:

$$\mathbb{P}(l) = \binom{n}{l}\binom{m-n}{d_2-l}\Big/\binom{m}{d_2}, 0 \le l \le d_2. \quad (30)$$

Then, the rest is to show that the subgraph $G^c(U, V_2, d_1, \bar{d})$ contains a perfect matching with high probability.

**Definition 13.** (Forbidden $k$-pair) *For a bipartite graph $G(U, V_2, d_1, \bar{d})$, a pair $(A, B)$ is called a $k$-blocking pair if $A \subseteq U$ with $|A| = k$, $B \subseteq V_2$ with $|B| = n - k + 1$, and there exists no arc between the nodes of sets $A$ and $B$. A blocking $k$-pair $(A, B)$ is called a forbidden pair if at least one of the following holds:*

1. $2 \le k < (n+1)/2$, *and for any $v_1 \in A$ and $v_2 \in V_2 \backslash B$, $(A\backslash\{v_1\}, B \cup \{v_2\})$ is not a $(k-1)$-blocking pair.*
2. $(n+1)/2 \le k \le n-1$, *and for any $v_1 \in U\backslash A$ and $v_2 \in B$, $(A \cup \{v_1\}, B\backslash\{v_2\})$ is not a $(k+1)$-blocking pair.*

The following technical lemma modified from [Walkup, 1980] is useful in our proof.

**Lemma 6.** *If the graph $G^c(U, V_2, d_1, \bar{d})$ does not contain a perfect matching, then there exists a forbidden $k$-pair for some $k$.*

*Proof.* One direct application of the Konig's theorem to bipartite graph shows that $G^c(U, V_2, d_1, \bar{d})$ contains a perfect matching if and only if it does not contain any blocking $k$-pair. It is rest to show that the existence of a $k$-blocking pair implies that there exists a forbidden $l-$pair for some $l$. Suppose that there exists a $k$-blocking pair $(A, B)$ with $k < (n+1)/2$, and it is not a forbidden $k$-pair. Otherwise, we already find a forbidden pair. Then, it implies that there exists $v_1 \in A$ and $v_2 \in V_2\backslash B$ such that $(A\backslash\{v_1\}, B\cup\{v_2\})$ is a $(k-1)$-blocking pair. Similarly, we can continue above argument on blocking pair $(A\backslash\{v_1\}, B \cup \{v_2\})$ until we find a forbidden pair. Otherwise, we will find a $1-$blocking pair $(A', B')$, which is a contradiction to our assumption that each node $v_1 \in U$ connects $d_1$ nodes in $V_2$. The proof for $k \ge (n+1)/2$ is same. □

Let $E$ be the event that graph $G(U, V_2, d_1, \bar{d})$ contains perfect matching. Based on the the results of Lemma 6, we have

$$\begin{aligned}
1 - \mathbb{P}(E) &= \mathbb{P}\left(\bigcup_{k=2}^{n-1} k\text{-forbidden pair exists}\right) \\
&\le \sum_{k=2}^{n-1} \mathbb{P}\left(k\text{-forbidden pair exists}\right) \\
&\le \sum_{k=2}^{n-1} \binom{n}{k}\binom{n}{n-k+1} \cdot \mathbb{P}\left((A,B) \text{ is } k\text{-forbidden pair}\right) \\
&= \sum_{k=2}^{n-1} \binom{n}{k}\binom{n}{n-k+1}\alpha(k)\beta(k).
\end{aligned}$$

The above, $A$ and $B$ are defined as node sets such that $A \subseteq U$ with $|A| = k$ and $B \subseteq V_2$ with $|B| = n - k + 1$. The $\alpha(k)$ and $\beta(k)$ are defined as follows.

$$\alpha(k) = \mathbb{P}\left((A,B) \text{ is } k\text{-forbidden pair} \middle| (A,B) \text{ is } k\text{-blocking pair}\right), \quad (31)$$

$$\beta(k) = \mathbb{P}\left((A,B) \text{ is } k\text{-blocking pair}\right). \quad (32)$$

From the Definition 13, it can be obtained the following estimation of probability $\beta(k)$.

$$\beta(k) = \left[\binom{k-1}{d_1}\Big/\binom{n}{d_1}\right]^k \cdot \left[\sum_{l=0}^{d_2}\binom{n-k}{l}\binom{m-n}{d_2-l}\Big/\binom{m}{d_2}\right]^{n-k+1}. \quad (33)$$

The first factor gives the probability that there exists no arc from nodes of $A$ to nodes of $B$. The second factor gives the probability that there exists no arc from nodes of $B$ to nodes of $A$. The summation operation in the second factor comes from conditioning such probability on the distribution (30). Based on the Chu-Vandermonde identity, one can simplify $\beta(k)$ as

$$\beta(k) = \left[\binom{k-1}{d_1}\Big/\binom{n}{d_1}\right]^k \cdot \left[\binom{m-k}{d_2}\Big/\binom{m}{d_2}\right]^{n-k+1}. \quad (34)$$

Utilizing the inequality

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n \le n! \le e^{\frac{1}{12n}}\sqrt{2\pi n}\left(\frac{n}{e}\right)^n, \quad (35)$$

we have

$$\binom{n}{k}\binom{n}{n-k+1} \le \frac{n^{2n}}{k^{2k}(n-k)^{2(n-k)}} \cdot \frac{ne^{1/6n}}{2\pi(n-k)(n-k+1)}, \quad (36)$$

$$\begin{aligned}
&\binom{k-1}{d_1}\Big/\binom{n}{d_1} \le c_1\sqrt{\frac{(k-1)(n-d_1)}{(k-d_1-1)n}} \cdot \\
&\left(\frac{k-1}{k-d_1-1}\right)^{k-d_1-1}\left(\frac{n-d_1}{n}\right)^{n-d_1}\left(\frac{k-1}{n}\right)^{d_1} \\
&\overset{(a)}{\le} c_1\sqrt{\frac{k-1}{k-d_1-1}}\left(\frac{n-d_1}{n}\right)^{\frac{1}{2}-d_1}\left(\frac{k-1}{n}\right)^{d_1}. \quad (37)
\end{aligned}$$

$$\binom{m-k}{d_2}\Big/\binom{m}{d_2} \overset{(a)}{\leq} c_2\sqrt{\frac{m-k}{m-d_2-k}}\left(\frac{m-d_2}{m}\right)^{\frac{1}{2}-d_2}\left(\frac{m-k}{m}\right)^{d_2}. \quad (38)$$

In the above, step (a) is based on fact that $(1+x/n)^n \leq e^x$ and parameters $c_1$ and $c_2$ are defined as

$$c_1 = e^{1/12(k-1)}e^{1/12(n-d_1)},\ c_2 = e^{1/12(m-k)}e^{1/12(m-d_2)}. \quad (39)$$

Combining the equations (36)-(39), we can obtain that

$$\begin{aligned}\gamma(k) =& \binom{n}{k}\binom{n}{n-k+1}\beta(k)\\ <& c_3\left(1-\frac{1}{k}\right)^{2k}\cdot\frac{n(m-k)^{d_2}}{m^{d_2}(n-k)(n-k+1)}\cdot\\ & \left[\frac{n^2(m-k)^{d_2}}{(n-k)^2m^{d_2}}\right]^{n-k}\cdot\left(\frac{k-1}{n}\right)^{(d_1-2)k}. \quad (40)\end{aligned}$$

The constant $c_3$ is given by $c_3 = e^{d_1^2+13d_1/12+d_2+1/3}/2\pi$. The third term satisfies, $\forall 2 \leq k \leq n-1$,

$$\frac{n^2(m-k)^{d_2}}{(n-k)^2m^{d_2}} \leq \max\left\{1, \frac{n^2(m-n+1)^{d_2}}{m^{d_2}}\right\}, \quad (41)$$

which is based on the fact that, if $d_2 > 2$, the function

$$f(k) = \frac{n^2(m-k)^{d_2}}{(n-k)^2m^{d_2}}$$

is monotonically decreasing when $k \leq (d_2n-2m)/(d_2-2)$ and increasing when $k \geq (d_2n-2m)/(d_2-2)$. If $d_2 = 2$, it is monotonically increasing for $k \geq 0$.

We then estimate the conditional probability $\alpha(k)$. Given a blocking pair $A \subseteq U$ with $|A| = k$ and $B \subseteq V_2$ with $|B| = n-k+1$, and a node $v_i \in A$, let $E_i$ be the set of nodes in $V_2\backslash B$ on which $d_1$ arcs from node $v_i$ terminate. Let $E'$ be the set of nodes $v$ in $V_2\backslash B$ such that at least 2 arcs leaving from $v$ to nodes in $A$. Then we have the following technical lemma.

**Lemma 7.** *Given a blocking pair $A \subseteq U$ with $|A| = k$ and $B \subseteq V_2$ with $|B| = n-k+1$, if $(A, B)$ is k-forbidden pair, then*

$$E^* = \left(\bigcup_{i=1}^{k} E_i\right) \cup E' = V_2\backslash B.$$

*Proof.* Suppose that there exists node $v \in V_2\backslash(E^* \cup B)$, then there exists no arc from $A$ to $v$ and there exists at most 1 arc from $v$ to $A$. If such an arc exists, let $v'$ be the corresponding terminating node in $A$. Then we have $(A\backslash\{v'\}, B\cup\{v\})$ is a blocking pair, which is contradictory to the definition of forbidden pair. If such an arc does not exist, let $v'$ be the an arbitrary node in $A$. Then we have $(A\backslash\{v'\}, B\cup\{v\})$ is a blocking pair, which is also contradictory to the definition of forbidden pair. □

The lemma 7 implies that we can upper bound the conditional probability by

$$\alpha(k) \leq \mathbb{P}\left[\left(\bigcup_{i=1}^{k} E_i\right) \cup E' = V_2\backslash B\right] = (1-P_1^kP_2)^{k-1},$$

where $P_1$ and $P_2$ is defined as: for any node $v \in V_2\backslash B$,

$$\begin{aligned}P_1 =& \mathbb{P}(v \notin E_i) = \binom{k-2}{d_1}\Big/\binom{k-1}{d_1} = \frac{k-d_1-1}{k-1},\\ P_2 =& \mathbb{P}(v \notin E')\\ =& 1-\sum_{l_1=2}^{d_2}\sum_{l_2=l_1}^{d_2}\mathbb{P}(l_2)\cdot\binom{k}{l_1}\binom{n-k}{l_2-l_1}\Big/\binom{n}{l_2}\\ \overset{(a)}{=}& \left[\binom{m-k}{d_2}+k\binom{m-k}{d_2-1}\right]\Big/\binom{m}{d_2}\\ \overset{(b)}{>}& e^{1/6}\left(\frac{m-k}{m}\right)^{m-k}\left(\frac{m-d_2}{m-d_2-k}\right)^{m-k-d_2}\left(\frac{m-d_2}{m}\right)^{k}\\ \overset{(c)}{>}& c_4e^{1/6-d_2}. \quad (42)\end{aligned}$$

The above, step (a) utilizes Chu-Vandermonde identity twice; step (b) is adopts the inequality (35); step (c) is based on the fact that if $n$ is sufficiently large, $(1-x/n)^n \geq c_5e^{-x}$, where $c_5$ is a constant. Combining the above estimation of $P_1$ and $P_2$, we have the following upper bound of $\alpha(k)$.

$$\alpha(k) < \left[1-c_6\left(\frac{k-d_1-1}{k-1}\right)^k\right]^{k-1}. \quad (43)$$

We finally estimate the probability that the graph $G^c(U, V_2, d_1, \bar{d})$ contains a perfect matching under the following two cases.

**Case 1:** The number of stragglers $s = \text{poly}(\log(n))$. Let $d_1 = 2, d_2 = 3$. Based on the estimation (41), we have that, for $n$ sufficiently large,

$$\frac{n^2(m-k)^3}{(n-k)^2m^3} \leq \max\left\{1, \frac{n^2(s+1)^3}{(n+s)^3}\right\} \leq 1. \quad (44)$$

Combining the above results with the estimation of $\beta(k)$, we have

$$\gamma(k) \leq \frac{c_3e^{-2}}{n}, 2 \leq k \leq n-1. \quad (45)$$

Then we can obtain that

$$\begin{aligned}&\mathbb{P}(G(U, V_2, d_1, \bar{d}) \text{ contains perfect matching})\\ \geq& 1-\sum_{k=1}^{n}\binom{n}{k}\binom{n}{n-k+1}\alpha(k)\beta(k)\\ \geq& 1-\sum_{k=2}^{n-1}\frac{c_3e^{-2}}{n}\alpha(k)\\ \overset{(a)}{>}& 1-\frac{c_3e^{-2}}{n}\sum_{k=2}^{n-1}\left[1-c_6\left(\frac{k-3}{k-1}\right)^k\right]^{k-1}\end{aligned}$$

$$\overset{(b)}{>} 1 - \frac{c_7}{n}. \tag{46}$$

The above, step (a) utilizes the estimation of $\alpha(k)$ in (43); step (b) is based on estimating the tail of the summation as geometric series $(1 - c_6/e^2)^{k-1}$.

**Case 2:** The number of stragglers $s = \Theta(n^\alpha)$, $\alpha < 1$. Let $d_1 = 2, d_2 = 2/(1-\alpha)$. For $2 \le k \le n-2$, we have

$$\frac{n^2(m-k)^3}{(n-k)^2 m^3} \le \max\left\{1, \frac{n^2(n^\alpha+2)^{2/(1-\alpha)}}{4(n+n^\alpha)^{2/(1-\alpha)}}\right\} \le 1, \tag{47}$$

for $n$ sufficiently large. Combining the above results with the estimation of $\beta(k)$, we have

$$\gamma(k) \le \frac{c_3 e^{-2}}{n}, 2 \le k \le n-2. \tag{48}$$

For $k = n-1$, we have

$$\frac{n^2(m-k)^3}{(n-k)^2 m^3} \le \max\left\{1, \left(\frac{n+n^{1-\alpha}}{n+n^\alpha}\right)^{2/(1-\alpha)}\right\}. \tag{49}$$

If $\alpha \ge 1/2$, we can directly obtain that $\gamma(k) \le \frac{c_3 e^{-2}}{n}$. If $\alpha < 1/2$, we have that, for $n$ sufficiently large,

$$\gamma(k) \le \frac{c_3 e^{-2}}{n}\left(\frac{n+n^{1-\alpha}}{n+n^\alpha}\right)^{4/(1-\alpha)} \le \frac{c_8 e^{-2}}{n}. \tag{50}$$

Similarly, we can obtain that

$$\begin{aligned}
&\mathbb{P}(G(U, V_2, d_1, \bar{d}) \text{ contains perfect matching}) \\
\ge& 1 - \sum_{k=1}^{n} \binom{n}{k}\binom{n}{n-k+1}\alpha(k)\beta(k) \\
\ge& 1 - \sum_{k=2}^{n-1} \frac{e^{-2}\max\{c_3, c_8\}}{n}\alpha(k) \\
\overset{(b)}{>}& 1 - \frac{c_9}{n}.
\end{aligned} \tag{51}$$

Therefore, in both cases, incorporating the above results into estimating (25), the theorem follows.

# H Experimental Results

In this section, we present the complete experimental results on Ohio Supercomputer Center Center [1987]. We compare our proposed coding schemes including the $s-$diagonal code and $(d_1, d_2)$-cross codes against the following existing schemes in both single matrix vector multiplication and gradient descent: (i) **uncoded scheme**: the input matrix is divided uniformly across all workers without replication and the master waits for all workers to send their results; (ii) **sparse MDS code** [Lee et al., 2017b]: the generator matrix is a sparse random Bernoulli matrix with average computation overhead $\Theta(\log(n))$. (iii) **polynomial code** [Yu et al., 2017]: coded matrix multiplication scheme with optimum recovery threshold and nearly linear decoding time; (iv) **short dot code** [Dutta et al., 2016]: append the dummy vectors to data matrix $\mathbf{A}$ before applying the MDS code, which provides some sparsity of encoded data matrix with cost of increased recovery threshold. (v) **LT code** [Luby, 2002]: rateless code widely used in broadcast communication. It achieves an average computation load of $\Theta(\log(n))$ and a nearly linear decoding time using peeling decoder. To simulate straggler effects in large-scale system, we randomly pick $s$ workers that are running a background thread.

## H.1 Coded Linear Transform

We implement all methods in python using MPI4py. Each worker stores the coded submatrix $\tilde{\mathbf{A}}_i$ according to the coding matrix $\mathbf{M}$. In the computation stage, each worker computes the linear transform $\tilde{\mathbf{A}}_i\mathbf{x}$ and returns the results using `Isend()`. Then the master node actively listens to the responses from each worker via `Irecv()`, and uses `Waitany()` to keep polling for the earliest finished tasks. Upon receiving enough results, the master stops listening and starts decoding the results.

We first use a matrix with $r = t = 1048576$ and $\text{nnz}(\mathbf{A}) = 89239674$ from data sets [Davis and Hu, 2011] , and evenly divide this matrix into $n = 12$ and 20 partitions. In Fig. 7 (a)(b), we report the job completion time under $s = 2$ and $s = 4$, based on 20 experimental runs. It can be observed that both $(2, 2)$-cross code outperforms uncoded scheme (in 50% the time), LT code (in 70% the time), sparse MDS code (in 60% the time), polynomial code (in 20% the time) and our $s$-diagonal code. Moreover, we observe that the uncoded scheme is faster than the polynomial code, because the input data matrix $\mathbf{A}$ is sparse and density of encoded data matrix is greatly increased, which leads to increased computation time per worker and additional I/O contention at the master node.

We further compare our proposed $s$-diagonal code with $(2, 2)$-cross code versus the number of stragglers $s$. As shown in Fig. 7(c)(d), when the number of stragglers $s$ increases, the job completion time of the $s$-diagonal code increases while the $(2, 2)$-cross code does not change. If the number of stragglers is smaller than 2, the $s$-diagonal code performs better than the $(2, 2)$-cross code. Another interesting observation is that the *irregularity of the work load* can decrease the I/O contention. For example, when $s = 2$, the computation load of the 2-diagonal code is similar as $(2, 2)$-cross code, which is equal to 36 in the case of $n = 12$. However, the $(2, 2)$-cross code costs less time due to the unbalanced worker load.

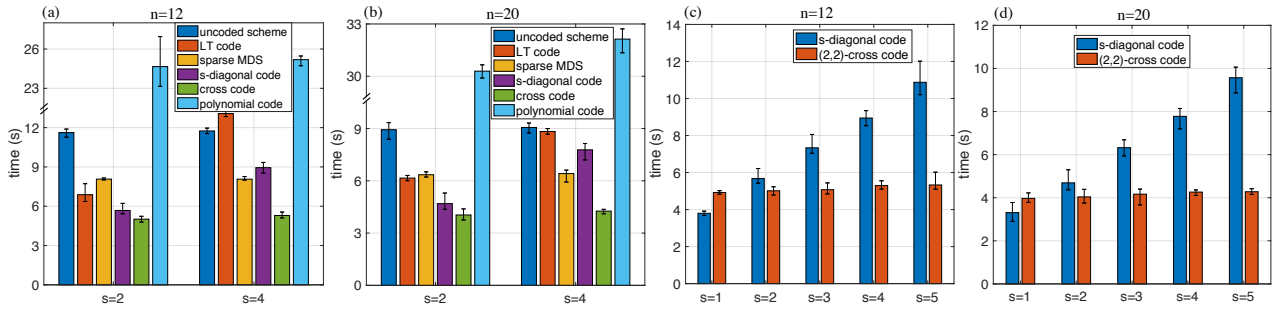


Figure 7: Comparison of job completion time including data transmission time, computation time and decoding time for $n = 12, 20$ and $s = 2, 4$.

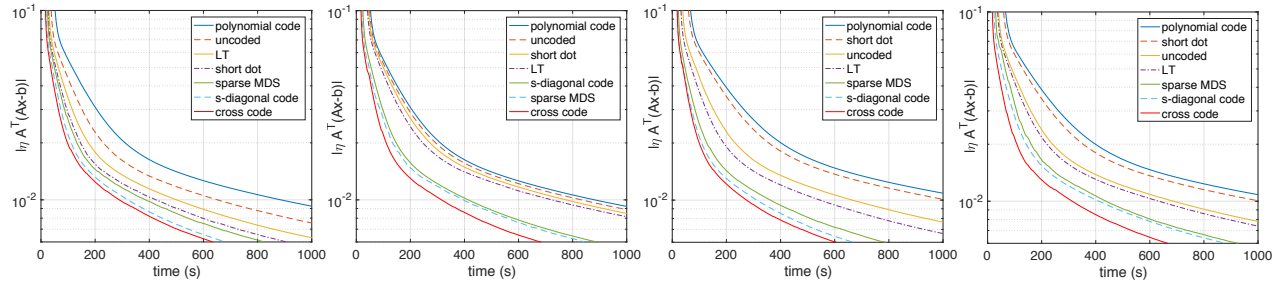


Figure 8: Magnitude of scaled gradient versus time for number of data partitions $n = 12$, 20 and number of stragglers $s = 2$,4.

### H.2 Coded Gradient Descent

We first describe the gradient-based distributed algorithm to solve the following linear regression problem.

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2, \tag{52}$$

where $\mathbf{A} \in \mathbb{R}^{r\times t}$ is the data matrix, $\mathbf{b} \in \mathbb{R}^r$ is the label vector and $\mathbf{x} \in \mathbb{R}^t$ is the unknown weight vector to be found. The standard gradient descent algorithm to solve the above problem is: in each iteration $t$,

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta\mathbf{A}^\intercal(\mathbf{A}\mathbf{x}_t - \mathbf{b}) = \mathbf{x}_t - \eta\sum_{i=1}^{n}\mathbf{A}_i^\intercal\mathbf{A}_i\mathbf{x}_t + \eta\mathbf{A}^\intercal\mathbf{b}. \tag{53}$$

In the uncoded gradient descent, each worker $i$ first stores a submatrix $\mathbf{A}_i$; then during iteration $t$, each worker $i$ first computes a vector $\mathbf{A}_i^\intercal\mathbf{A}_i\mathbf{x}_t$ and returns it to the master node; the master node then updates the weight vector $\mathbf{x}_t$ according to the above gradient descent step and assigns the new weight vector $\mathbf{x}_{t+1}$ to each worker. The algorithm terminates when the gradient vanishes, i.e., $\|\mathbf{A}^\intercal(\mathbf{A}\mathbf{x}_t - \mathbf{b})\| \le \epsilon$. In the coded gradient descent, each worker $i$ first stores several submatrices according to the coding matrix $\mathbf{M}$; during iteration $t$, each worker $i$ computes a linear combination,

$$\sum_{j=1}^{n} m_{ij}\mathbf{A}_j^\intercal\mathbf{A}_j\mathbf{x}_t, \forall i \in [m]. \tag{54}$$

where $m_{ij}$ is the element of the coding matrix $\mathbf{M}$. The master node collects a subset of results, decodes the full gradient $\mathbf{A}^\intercal(\mathbf{A}\mathbf{x}_t - \mathbf{b})$ and updates the weight vector $\mathbf{x}_t$. Then continue to the next round.

We use data from LIBSVM dataset repository with $r = 19264097$ samples and $t = 1163024$ features. We evenly divide the data matrix $\mathbf{A}$ into $n = 12, 20$ submatrices. In Fig. 8 , we plot the magnitude of scaled gradient $\|\eta\mathbf{A}^\intercal(\mathbf{A}\mathbf{x} - \mathbf{b})\|$ versus the running times of the above seven different schemes under $n = 12, 20$ and $s = 2, 4$. Among all experiments, we can see that the $(2, 2)$-cross code converges at least $30\%$ faster than sparse MDS code, 2 times faster than both uncoded scheme and LT code and at least 4 times faster than short dot and polynomial code. The $(2, 2)$-cross code performs similar with $s$-diagonal code when $s = 2$ and converges $30\%$ faster than $s$-diagonal code when $s = 4$.