A. Proofs

A.1. Proof of Theorem 3.1

For simplicity of proof, let us define a valid s-attack first.

Definition A.1. $\mathbf{N} = [\mathbf{n}_1, \mathbf{n}_2, \cdots, \mathbf{n}_P]$ *is a valid s attack if and only if* $|\{j : \|\mathbf{n}_j\|_0 \neq 0\}| \leq s$.

Now we prove theorem 3.1. Suppose $(\mathbf{A}, E, D)$ can resist s adversaries. The goal is to prove $\|A\|_0 \geq P(2s+1)$. In fact we can prove a slightly stronger version: $\|\mathbf{A}_{\cdot,i}\|_0 \geq (2s+1)\,, i = 1, 2, \cdots, B$. Suppose for some i, $\|\mathbf{A}_{\cdot,i}\|_0 = \tau < (2s+1)$. Without loss of generality, assume that $\mathbf{A}_{1,i}, \mathbf{A}_{2,i}, \mathbf{A}_{\tau,i}$ are non-zero. Let $\mathbf{G}_{-i} = [\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_{i-1}, \mathbf{g}_{i+1}, \cdots, \mathbf{g}_P]$. Since $(\mathbf{A}, E, D)$ can protect against s adversaries, we have for any $\mathbf{G}$,

$$D(\mathbf{Z}^{\mathbf{A},E,\mathbf{G}} + \mathbf{N}) = \mathbf{G}\mathbf{1} = \mathbf{G}_{-i}\mathbf{1} + \mathbf{g}_i,$$

for any valid s-attack $\mathbf{N}$. In particular, let $\mathbf{g}_i^1 = \mathbf{1}_d$, $\mathbf{g}_2^i = -\mathbf{1}_d$, $\mathbf{G}^1 = [\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_{i-1}, \mathbf{g}_i^1, \mathbf{g}_{i+1}, \cdots, \mathbf{g}_P]$, and $\mathbf{G}^2 = [\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_{i-1}, \mathbf{g}_i^2, \mathbf{g}_{i+1}, \cdots, \mathbf{g}_P]$. Then for any valid s attack $\mathbf{N}^1, \mathbf{N}^2$,

$$D(\mathbf{Z}^{\mathbf{A},E,\mathbf{G}^1} + \mathbf{N}^1) = \mathbf{G}_{-i}\mathbf{1}_{P-1} + \mathbf{1}_d.$$

and

$$D(\mathbf{Z}^{\mathbf{A},E,\mathbf{G}^2} + \mathbf{N}^2) = \mathbf{G}_{-i}\mathbf{1}_{P-1} - \mathbf{1}_d.$$

Our goal is to find $\mathbf{N}^1, \mathbf{N}^2$ such that $D(\mathbf{Z}^{\mathbf{A},E,\mathbf{G}^1} + \mathbf{N}^1) = D(\mathbf{Z}^{\mathbf{A},E,\mathbf{G}^2} + \mathbf{N}^2)$ which then will lead to a contradiction. Construct $\mathbf{N}^1$ and $\mathbf{N}^2$ by

$$\mathbf{N}^1_{\ell,j} = \begin{cases} \left[\mathbf{Z}^{\mathbf{A},E,\mathbf{N}^2}\right]_{\ell,j} - \left[\mathbf{Z}^{\mathbf{A},E,\mathbf{N}^1}\right]_{\ell,j}, & j = 1, 2, \cdots, \lceil\frac{\tau-1}{2}\rceil \\ 0, & \text{otherwise} \end{cases}$$

and

$$\mathbf{N}^2_{\ell,j} = \begin{cases} \left[\mathbf{Z}^{\mathbf{A},E,\mathbf{N}^1}\right]_{\ell,j} - \left[\mathbf{Z}^{\mathbf{A},E,\mathbf{N}^2}\right]_{\ell,j}, & j = \lceil\frac{\tau-1}{2}\rceil, \lceil\frac{\tau-1}{2}\rceil + 1, \cdots, \tau \\ 0, & \text{otherwise} \end{cases}$$

One can easily verify that $\mathbf{N}^1, \mathbf{N}^2$ are both valid s attack. Meanwhile, we have

$$\left[\mathbf{Z}^{\mathbf{A},E,\mathbf{G}^1}\right]_{\ell,j} + \mathbf{N}^1_{\ell,j} = \left[\mathbf{Z}^{\mathbf{A},E,\mathbf{G}^2}\right]_{\ell,j} + \mathbf{N}^2_{\ell,j}, j = 1, 2, \cdots, \tau$$

due to the above construction of $\mathbf{N}^1, \mathbf{N}^2$. Note that $\mathbf{A}_{j,i} = 0$ for all $j > \tau$, which implies that for all compute nodes with index $j > \tau$, their encoder functions do not depend on the ith gradient. Since $\mathbf{G}^1$ and $\mathbf{G}^2$ only differ in the ith gradient, the encoder function of any compute node with index $j > \tau$ should have the same output. Thus, we have

$$\left[\mathbf{Z}^{\mathbf{A},E,\mathbf{G}^1}\right]_{\ell,j} + \mathbf{N}^1_{\ell,j} = \left[\mathbf{Z}^{\mathbf{A},E,\mathbf{G}^1}\right]_{\ell,j} = \left[\mathbf{Z}^{\mathbf{A},E,\mathbf{G}^2}\right]_{\ell,j} = \left[\mathbf{Z}^{\mathbf{A},E,\mathbf{G}^2}\right]_{\ell,j} + \mathbf{N}^2_{\ell,j}, j > \tau$$

Hence, we have

$$\left[\mathbf{Z}^{\mathbf{A},E,\mathbf{G}^1}\right]_{\ell,j} + \mathbf{N}^1_{\ell,j} = \left[\mathbf{Z}^{\mathbf{A},E,\mathbf{G}^2}\right]_{\ell,j} + \mathbf{N}^2_{\ell,j}, \forall j$$

which means

$$\mathbf{Z}^{\mathbf{A},E,\mathbf{G}^1} + \mathbf{N}^1 = \mathbf{Z}^{\mathbf{A},E,\mathbf{G}^2} + \mathbf{N}^2$$

Therefore, we have

$$D(\mathbf{Z}^{\mathbf{A},E,\mathbf{G}^1} + \mathbf{N}^1) = D(\mathbf{Z}^{\mathbf{A},E,\mathbf{G}^2} + \mathbf{N}^2)$$

and thus

$$\mathbf{G}_{-1}\mathbf{1}_{P-1} + \mathbf{1}_d = D(\mathbf{Z}^{\mathbf{A},E,\mathbf{G}^1} + \mathbf{N}^1) = D(\mathbf{Z}^{\mathbf{A},E,\mathbf{G}^2} + \mathbf{N}^2) = \mathbf{G}_{-1}\mathbf{1}_{P-1} - \mathbf{1}_d$$

This gives us a contradiction. Hence, the assumption is not correct and we must have $\|\mathbf{A}_{\cdot,i}\|_0 \geq (2s+1)\,, i = 1, 2, \cdots, P$. Thus, we must have $\|A\|_0 \geq (2s+1)P$. □

A direct but interesting corollary of this theorem is a bound on the number of adversaries DRACO can resist.

Corollary A.1. *$(\mathbf{A}, E, D)$ can resist at most $\frac{P-1}{2}$ adversarial nodes.*

Proof. According to Theorem 3.1, the redundancy ratio is at least $2s+1$, meaning that every data point must be replicated by at least $2s+1$. Since there are P compute node in total, we must have $2s+1 \leq P$, which implies $s \leq \frac{P-1}{2}$. Thus, $(\mathbf{A}, E, D)$ can resist at most $\frac{P-1}{2}$ adversaries. □

In other words, at least a majority of the compute nodes must be non-adversarial. □

A.2. Proof of Theorem 3.2

Since there are at most s adversaries, there are at least $2s+1-s = s+1$ non-adversarial compute nodes in each group. Thus, performing majority vote on each group returns the correct gradient, and thus the repetition code guarantees that the result is correct. The complexity at each compute node is clearly $\mathcal{O}((2s+1)d)$ since each of them only computes the sum of $(2s+1)$ d-dimensional gradients. For the decoder at the PS, within each group of $(2s+1)$ machine, it takes $\mathcal{O}((2s+1)d)$ computations to find the majority. Since there are $\frac{P}{(2s+1)}$ groups, it takes in total $\mathcal{O}((2s+1)d\frac{P}{(2s+1)}) = \mathcal{O}(Pd)$ computations. Thus, this achieves linear-time encoding and decoding. □

A.3. Proof of Lemma 3.3

We first prove that $\mathbf{A}_{j,k} = 0 \Rightarrow \mathbf{W}_{j,k} = 0$.

Suppose $\mathbf{A}_{j,k} = 0$ for some j, k. Then by definition $k \in \alpha_j$. By $\mathbf{0} = \begin{bmatrix}\mathbf{q}_j & 1\end{bmatrix} \cdot [\mathbf{C}_L]_{\cdot,\alpha_j}$ we have $0 = \begin{bmatrix}\mathbf{q}_j & 1\end{bmatrix} [\mathbf{C}_L]_{\cdot,k} = \mathbf{W}_{j,k}$.

Next we prove that for any index set U such that $|U| \geq P - (2s+1)$, the column span of $\mathbf{W}_{\cdot,U}$ contains $\mathbf{1}$. This is equivalent to that for any index set U such that $|U| \geq P - (2s+1)$, there exists a vector $\mathbf{b}$ such that $\mathbf{W}_{\cdot,U}b = \mathbf{1}$. Now we show such b exists. Note that $\mathbf{C}_L$ is a $(P-2s) \times P$ full rank Vandermonde matrix and thus any $P-2s$ columns of $\mathbf{C}_L$ are linearly independent. Let $\bar{U}$ be the first $P-2s$ elements in U. Then all columns of $[\mathbf{C}_L]_{\cdot,\bar{U}}$ are linearly independent and thus $[\mathbf{C}_L]_{\cdot,\bar{U}}$ is invertible. Let $\mathbf{b}_{\bar{U}} \triangleq \bar{\mathbf{b}} = \left(C^L_{\bar{U}}\right)^{-1} \begin{bmatrix}0 & 0 & \cdots & 0 & 1\end{bmatrix}^T$. For any $j \notin \bar{U}$, let $\mathbf{b}_j = 0$. Then we have

$$\begin{aligned}
\mathbf{W}_U\mathbf{b} &= \begin{bmatrix}\mathbf{Q} & \mathbf{1}\end{bmatrix} \times [\mathbf{C}_L]_{\cdot,U}\,\mathbf{b} \\
&= \begin{bmatrix}\mathbf{Q} & \mathbf{1}\end{bmatrix} \times [\mathbf{C}_L]_{\cdot,\bar{U}}\,\bar{\mathbf{b}} \\
&= \begin{bmatrix}\mathbf{Q} & \mathbf{1}\end{bmatrix} \times [\mathbf{C}_L]_{\cdot,\bar{U}} \times [\mathbf{C}_L]^{-1}_{\cdot,\bar{U}} \begin{bmatrix}0 & 0 & \cdots & 0 & 1\end{bmatrix}^T \\
&= \begin{bmatrix}\mathbf{Q} & \mathbf{1}\end{bmatrix}\begin{bmatrix}0 & 0 & \cdots & 0 & 1\end{bmatrix}^T \\
&= \mathbf{1}.
\end{aligned}$$

This completes the proof. □

A.4. Proof of Lemma 3.4

We need a few lemmas first.

Lemma A.2. *Let a P-dimensional vector $\gamma \triangleq [\gamma_1, \gamma_2, \cdots, \gamma_P]^T = (\mathbf{f}\mathbf{N})^T$. Then we have*

$$Pr(\{j : \gamma_j \neq 0\} = \{j : \|\mathbf{N}_{\cdot,j}\|_0 \neq 0\}) = 1.$$

Proof. Let us prove that

$$Pr(\mathbf{N}_{\cdot,j} \neq 0\}|\gamma_j \neq 0) = 1.$$

and

$$Pr(\gamma_j \neq 0|\mathbf{N}_{\cdot,j} \neq 0\}) = 1.$$

for any j. Combining those two equations we prove the lemma.

The first equation is straightforward. Suppose $\mathbf{N}_{\cdot,j} = 0$. Then we immediately have $\gamma_j = \mathbf{f}\mathbf{N}_{\cdot,j} = 0$. For the second one, note that $\mathbf{f}$ has entries drawn independently from the standard normal distribution. Therefore we have that $\gamma_j = \mathbf{f}\mathbf{N}_{\cdot,j} \sim \mathcal{N}(\mathbf{1}^T\mathbf{N}_{\cdot,j}, \|\mathbf{N}_{\cdot,j}\|_2^2)$. Since γ_j is a random variable with normal distribution, the probability of it being any particular value is 0. In particular,

$$Pr(\gamma_j = 0|\mathbf{N}_{\cdot,j} \neq 0\}) = 0,$$

and thus

$$Pr(\gamma_j \neq 0|\mathbf{N}_{\cdot,j} \neq 0\}) = 1$$

which proves the second equation and finishes the proof. □

Lemma A.3. $\mathbf{R}^{Cyc}\mathbf{C}_R^\dagger = \mathbf{N}\mathbf{C}_R^\dagger$.

Proof. By definition, $\mathbf{R}^{Cyc}\mathbf{C}_R^\dagger = (\mathbf{G}\mathbf{W} + \mathbf{N})\,\mathbf{C}_R^\dagger = \big(\mathbf{G}\big[\mathbf{Q} \quad \mathbf{1}\big]\mathbf{C}_L + \mathbf{N}\big)\,\mathbf{C}_R^\dagger = \mathbf{G}\big[\mathbf{Q} \quad \mathbf{1}\big]\mathbf{C}_L\mathbf{C}_R^\dagger + \mathbf{N}\mathbf{C}_R^\dagger = \mathbf{N}\mathbf{C}_R^\dagger$. In the last equation we use the fact that IDFT matrix is unitary and thus $\mathbf{C}_L\mathbf{C}_R^\dagger = \mathbf{0}_{(P-2s)\times(2s)}$. □

Lemma A.4. *Let a P-dimensional vector $\hat{\mathbf{h}} \triangleq [\hat{h}_0, \hat{h}_1, \cdots, \hat{h}_{P-1}]^T$ be the discrete Fourier transformation (DFT) of a P-dimensional vector $\hat{\mathbf{t}} \triangleq [\hat{t}_1, \hat{t}_2, \cdots, \hat{t}_{P-1}]^T$ which has at most s non-zero elements, i.e., $\hat{\mathbf{h}} = \mathbf{C}^\dagger\hat{\mathbf{t}}$ and $\|\mathbf{t}\|_0 \leq s$. Then there exists a s-dimensional vector $\hat{\beta} \triangleq [\hat{\beta}_0, \hat{\beta}_1, \cdots, \hat{\beta}_{s-1}]^T$, such that*

$$\begin{bmatrix} \hat{h}_{P-s-1} & \hat{h}_{P-s} & \dots & \hat{h}_{P-2} \\ \hat{h}_{P-s-2} & \hat{h}_{P-s-1} & \dots & \hat{h}_{P-3} \\ \dots & \dots & \ddots & \vdots \\ \hat{h}_{P-2s} & \hat{h}_{P-s+1} & \dots & \hat{h}_{P-s-1} \end{bmatrix} \hat{\beta} = \begin{bmatrix} \hat{h}_{P-1} \\ \hat{h}_{P-2} \\ \vdots \\ \hat{h}_{P-s} \end{bmatrix}. \tag{A.1}$$

Furthermore, for any $\hat{\beta}$ satisfying the above equations,

$$\hat{h}_\ell = \sum_{u=0}^{s-1} \hat{\beta}_u \hat{h}_{\ell+u-s}, \tag{A.2}$$

always holds for all ℓ, where $\hat{h}_\ell = \hat{h}_{P+\ell}$.

Proof. Let $i_1, i_2, \cdots, i_s$ be the index of the non-zero elements in $\hat{\mathbf{t}}$. Let us define the location polynomial $p(\omega) = \prod_{k=1}^{s}(\omega - e^{-\frac{2\pi i}{P}i_k}) \triangleq \sum_{k=0}^{s}\theta_k\omega^k$, where $\theta_s = 1$. Let a s-dimensional vector $\hat{\beta}^* \triangleq -[\theta_0, \theta_1, \cdots, \theta_{s-1}]^T$.

Now we prove that $\hat{\beta} = \hat{\beta}^*$ is a solution to the system of linear equations (A.1). To see this, note that by definition, for any λ, we have $0 = p(e^{-\frac{2\pi i}{P}i_\lambda}) = \sum_{k=0}^{s}\theta_k e^{-\frac{2\pi i}{P}i_\lambda k}$. Multiply both side by $\hat{t}_{i_\lambda}e^{-\frac{2\pi i}{P}i_\lambda\eta}$, we have

$$\begin{aligned} 0 &= \hat{t}_{i_\lambda}e^{-\frac{2\pi i}{P}i_\lambda\eta}\sum_{k=0}^{s}\theta_k e^{-\frac{2\pi i}{P}i_\lambda k} \\ &= \hat{t}_{i_\lambda}\sum_{k=0}^{s}\theta_k e^{-\frac{2\pi i}{P}i_\lambda(k+\eta)}. \end{aligned}$$

Summing over λ, we have

$$\begin{aligned} 0 &= \sum_{\lambda=1}^{s} \hat{t}_{i_\lambda} \sum_{k=0}^{s} \theta_k e^{-\frac{2\pi i}{P} i_\lambda (k+\eta)} \\ &= \sum_{k=0}^{s} \theta_k \sum_{\lambda=1}^{s} \hat{t}_{i_\lambda} e^{-\frac{2\pi i}{P} i_\lambda (k+\eta)}. \end{aligned}$$

By definition, $\hat{h}_j = \mathbf{C}_{j,\cdot}\hat{\mathbf{t}} = \frac{1}{\sqrt{P}} \sum_{k=0}^{P-1} e^{-\frac{2\pi i}{P} jk} \hat{t}_k = \frac{1}{\sqrt{P}} \sum_{\lambda=1}^{s} \hat{t}_{i_\lambda} e^{-\frac{2\pi i}{P} i_\lambda j}$. Hence, the above equation becomes

$$0 = \sum_{k=0}^{s} \theta_k \sqrt{P} \hat{h}_{k+\eta}$$

which is equivalent to

$$\hat{h}_{s+\eta} = \sum_{k=0}^{s-1} -\theta_k \hat{h}_{k+\eta}$$

due to the fact that $\theta_s = 1$. By setting $\eta = -s + P - 1, -s + P - 2, \cdots, -s + P - s$, one can easily see that the above equation becomes identical to the system of linear equations in (A.1) with $\hat{\beta} = \hat{\beta}^* = -[\theta_0, \theta_1, \cdots, \theta_{s-1}]^T$.

Now let us prove for any $\hat{\beta}$ that satisfies equation (A.1), we have (A.2). Note that an equivalent form of (A.2) is that the following system of linear equations

$$\begin{bmatrix} \hat{h}_{P-s-1+\ell} & \hat{h}_{P-s+\ell} & \cdots & \hat{h}_{P-2+\ell} \\ \hat{h}_{P-s-2+\ell} & \hat{h}_{P-s-1+\ell} & \cdots & \hat{h}_{P-3+\ell} \\ \cdots & \cdots & \ddots & \vdots \\ \hat{h}_{P-2s+\ell} & \hat{h}_{P-s+1+\ell} & \cdots & \hat{h}_{P-s-1+\ell} \end{bmatrix} \hat{\beta} = \begin{bmatrix} \hat{h}_{P-1+\ell} \\ \hat{h}_{P-2+\ell} \\ \vdots \\ \hat{h}_{P-s+\ell} \end{bmatrix} \tag{A.3}$$

holds for $\ell = 0, 1, 2 \cdots, P - 1$. We prove this by induction. When $\ell = 1$, this is true since $\hat{\beta}$ satisfies the system of linear equations in (A.1). Assume it holds for $\ell = \mu$, i.e.,

$$\begin{bmatrix} \hat{h}_{P-s-1+\mu} & \hat{h}_{P-s+\mu} & \cdots & \hat{h}_{P-2+\mu} \\ \hat{h}_{P-s-2+\mu} & \hat{h}_{P-s-1+\mu} & \cdots & \hat{h}_{P-3+\mu} \\ \cdots & \cdots & \ddots & \vdots \\ \hat{h}_{P-2s+\mu} & \hat{h}_{P-s+1+\mu} & \cdots & \hat{h}_{P-s-1+\mu} \end{bmatrix} \hat{\beta} = \begin{bmatrix} \hat{h}_{P-1+\mu} \\ \hat{h}_{P-2+\mu} \\ \vdots \\ \hat{h}_{P-s+\mu} \end{bmatrix}$$

Now we need to prove it also holds when $\ell = \mu + 1$, *i*.e.,

$$\begin{bmatrix} \hat{h}_{P-s-1+\mu+1} & \hat{h}_{P-s+\mu+1} & \cdots & \hat{h}_{P-2+\mu+1} \\ \hat{h}_{P-s-2+\mu+1} & \hat{h}_{P-s-1+\mu+1} & \cdots & \hat{h}_{P-3+\mu+1} \\ \cdots & \cdots & \ddots & \vdots \\ \hat{h}_{P-2s+\mu+1} & \hat{h}_{P-s+1+\mu+1} & \cdots & \hat{h}_{P-s-1+\mu+1} \end{bmatrix} \hat{\beta} = \begin{bmatrix} \hat{h}_{P-1+\mu+1} \\ \hat{h}_{P-2+\mu+1} \\ \vdots \\ \hat{h}_{P-s+\mu+1} \end{bmatrix}.$$

First, since both $\hat{\beta}, \hat{\beta}^*$ satisfy the induction assumption, we must have

$$\begin{bmatrix} \hat{h}_{P-s-1+\mu} & \hat{h}_{P-s+\mu} & \cdots & \hat{h}_{P-2+\mu} \\ \hat{h}_{P-s-2+\mu} & \hat{h}_{P-s-1+\mu} & \cdots & \hat{h}_{P-3+\mu} \\ \cdots & \cdots & \ddots & \vdots \\ \hat{h}_{P-2s+\mu} & \hat{h}_{P-s+1+\mu} & \cdots & \hat{h}_{P-s-1+\mu} \end{bmatrix} (\hat{\beta} - \hat{\beta}^*) = \mathbf{0}_s.$$

Due to the induction assumption, one can verify that

$$[\theta_{s-1},\theta_{s-2},\cdots,\theta_0]\begin{bmatrix}\hat{h}_{P-s-1+\mu} & \hat{h}_{P-s+\mu} & \cdots & \hat{h}_{P-2+\mu}\\ \hat{h}_{P-s-2+\mu} & \hat{h}_{P-s-1+\mu} & \cdots & \hat{h}_{P-3+\mu}\\ \cdots & \cdots & \ddots & \vdots\\ \hat{h}_{P-2s+\mu} & \hat{h}_{P-s+1+\mu} & \cdots & \hat{h}_{P-s-1+\mu}\end{bmatrix} = \begin{bmatrix}\hat{h}_{P-s+\mu} & \hat{h}_{P-s+\mu+1} & \cdots\hat{h}_{P-2+\mu+1}\end{bmatrix},$$

and thus we have

$$\begin{aligned}&\begin{bmatrix}\hat{h}_{P-s+\mu} & \hat{h}_{P-s+\mu+1} & \cdots\hat{h}_{P-2+\mu+1}\end{bmatrix}(\hat{\beta}-\hat{\beta}^*)\\ =&[\theta_{s-1},\theta_{s-2},\cdots,\theta_0]\begin{bmatrix}\hat{h}_{P-s-1+\mu} & \hat{h}_{P-s+\mu} & \cdots & \hat{h}_{P-2+\mu}\\ \hat{h}_{P-s-2+\mu} & \hat{h}_{P-s-1+\mu} & \cdots & \hat{h}_{P-3+\mu}\\ \cdots & \cdots & \ddots & \vdots\\ \hat{h}_{P-2s+\mu} & \hat{h}_{P-s+1+\mu} & \cdots & \hat{h}_{P-s-1+\mu}\end{bmatrix}(\hat{\beta}-\hat{\beta}^*)=0.\end{aligned}$$

Hence,

$$\begin{aligned}&\begin{bmatrix}\hat{h}_{P-s+\mu} & \hat{h}_{P-s-1+\mu} & \cdots & \hat{h}_{P-1+\mu}\end{bmatrix}\hat{\beta}\\ =&\begin{bmatrix}\hat{h}_{P-s+\mu} & \hat{h}_{P-s-1+\mu} & \cdots & \hat{h}_{P-1+\mu}\end{bmatrix}\hat{\beta}^* + \begin{bmatrix}\hat{h}_{P-s+\mu} & \hat{h}_{P-s-1+\mu} & \cdots & \hat{h}_{P-1+\mu}\end{bmatrix}(\hat{\beta}-\hat{\beta}^*) = \hat{h}_{P+\mu} = \hat{h}_{P-1+\mu+1}.\end{aligned}$$

Furthermore, by induction assumption, we have

$$\begin{aligned}&\begin{bmatrix}\hat{h}_{P-s-2+\mu+1} & \hat{h}_{P-s-1+\mu+1} & \cdots & \hat{h}_{P-3+\mu+1}\\ \hat{h}_{P-s-3+\mu+1} & \hat{h}_{P-s-2+\mu+1} & \cdots & \hat{h}_{P-4+\mu+1}\\ \cdots & \cdots & \ddots & \vdots\\ \hat{h}_{P-2s+\mu+1} & \hat{h}_{P-s+1+\mu+1} & \cdots & \hat{h}_{P-s+1+\mu+1}\end{bmatrix}\hat{\beta} = \begin{bmatrix}\hat{h}_{P-s-1+\mu} & \hat{h}_{P-s-2+\mu} & \cdots & \hat{h}_{P-2+\mu}\\ \hat{h}_{P-s-2+\mu} & \hat{h}_{P-s-1+\mu} & \cdots & \hat{h}_{P-3+\mu}\\ \cdots & \cdots & \ddots & \vdots\\ \hat{h}_{P-(2s-1)+\mu} & \hat{h}_{P-s+\mu} & \cdots & \hat{h}_{P-s+\mu}\end{bmatrix}\hat{\beta}\\ =&\begin{bmatrix}\hat{h}_{P-1+\mu}\\ \hat{h}_{P-2+\mu}\\ \vdots\\ \hat{h}_{P-(s-1)+\mu}\end{bmatrix} = \begin{bmatrix}\hat{h}_{P-2+(\mu+1)}\\ \hat{h}_{P-3+(\mu+1)}\\ \vdots\\ \hat{h}_{P-s+(\mu+1)}\end{bmatrix}.\end{aligned}$$

Combing those two result we have proved

$$\begin{bmatrix}\hat{h}_{P-s-1+\mu+1} & \hat{h}_{P-s+\mu+1} & \cdots & \hat{h}_{P-2+\mu+1}\\ \hat{h}_{P-s-2+\mu+1} & \hat{h}_{P-s-1+\mu+1} & \cdots & \hat{h}_{P-3+\mu+1}\\ \cdots & \cdots & \ddots & \vdots\\ \hat{h}_{P-2s+\mu+1} & \hat{h}_{P-s+1+\mu+1} & \cdots & \hat{h}_{P-s-1+\mu+1}\end{bmatrix}\hat{\beta} = \begin{bmatrix}\hat{h}_{P-1+\mu+1}\\ \hat{h}_{P-2+\mu+1}\\ \vdots\\ \hat{h}_{P-s+\mu+1}\end{bmatrix}.$$

By induction, the equation A.3 holds for all $\ell = 0, 1, \cdots, P-1$. Equation A.3 immediately finishes the proof. □

Now we are ready to prove Lemma 3.4. By Lemma A.2, for the P-dimensional vector $\gamma = (\mathbf{f}\mathbf{N})^T$, we have

$$Pr(\{j : \gamma_j \neq 0\} = \{j : \|\mathbf{N}_{\cdot,j}\|_0 \neq 0\}) = 1,$$

Since there are at most s adversaries, the number of non-zero columns in $\mathbf{N}$ is at most s and hence there are at most s non-zero elements in γ, i.e., $\|\gamma\|_0 \leq s$, with probability 1. Now consider the case when $\|\gamma\|_0 \leq s$. First note that $[h_{P-2s}, h_{P-2s+1}, \cdots, h_{P-1}] = \mathbf{f}\mathbf{R}^{Cyc}\mathbf{C}_R^\dagger = \mathbf{f}\mathbf{N}\mathbf{C}_R^\dagger = \gamma^T\mathbf{C}_R^\dagger$, where the second equation is due to Lemma A.3. Now let us construct $\hat{\mathbf{h}} = [\hat{h}_0, \hat{h}_1, \cdots, \hat{h}_{P-1}]^T$ by $\hat{\mathbf{h}} = \mathbf{C}^\dagger\gamma$. Note that $\mathbf{C}$ is symmetric and thus $\mathbf{C}^\dagger = \left[\mathbf{C}^\dagger\right]^T$. One can easily verify that $\hat{h}_\ell = h_\ell, \ell = P-2s, P-2s+1, \cdots, P-1$. Therefore, the equation

$$\begin{bmatrix}h_{P-s-1} & h_{P-s} & \cdots & h_{P-2}\\ h_{P-s-2} & h_{P-s-1} & \cdots & h_{P-3}\\ \cdots & \cdots & \ddots & \vdots\\ h_{P-2s} & h_{P-s+1} & \cdots & h_{P-s+1}\end{bmatrix}\begin{bmatrix}\beta_0\\ \beta_1\\ \vdots\\ \beta_{s-1}\end{bmatrix} = \begin{bmatrix}h_{P-1}\\ h_{P-2}\\ \vdots\\ h_{P-s}\end{bmatrix}$$

becomes

$$\begin{bmatrix} \hat{h}_{P-s-1} & h_{P-s} & \dots & \hat{h}_{P-2} \\ \hat{h}_{P-s-2} & \hat{h}_{P-s-1} & \dots & \hat{h}_{P-3} \\ \dots & \dots & \ddots & \vdots \\ \hat{h}_{P-2s} & \hat{h}_{P-s+1} & \dots & \hat{h}_{P-s+1} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{s-1} \end{bmatrix} = \begin{bmatrix} \hat{h}_{P-1} \\ \hat{h}_{P-2} \\ \vdots \\ \hat{h}_{P-s} \end{bmatrix}$$

which always has a solution. Assume we find one solution $\bar{\beta} = [\bar{\beta}_0, \bar{\beta}_1, \cdots, \bar{\beta}_{P-1}]^T$. By the second part of Lemma A.4, we have

$$\hat{h}_\ell = \sum_{u=0}^{s-1} \bar{\beta}_u \hat{h}_{\ell+u-s}, \forall \ell.$$

Now we prove by induction that $h_\ell = \hat{h}_\ell, \ell = 0, 1, \cdots, P-1$.

When $\ell = 0$, we have

$$\hat{h}_0 = \sum_{u=0}^{s-1} \bar{\beta}_u \hat{h}_{u-s} = \sum_{u=0}^{s-1} \bar{\beta}_u h_{u-s} = h_0$$

where the second equation is due to the fact that $[h_{P-2s}, h_{P-2s-1}, \cdots, h_{P-1}] = [\hat{h}_{P-2s}, \hat{h}_{P-2s-1}, \cdots, \hat{h}_{P-1}]$ and $\hat{h}_{P+\ell} = \hat{h}_\ell, h_{P+\ell} = h_\ell$ (by definition).

Assume that for $\ell \leq \mu$, $\hat{h}_\ell = h_\ell$.

When $\ell = \mu + 1$, we have

$$\hat{h}_{\mu+1} = \sum_{u=0}^{s-1} \bar{\beta}_u \hat{h}_{\mu+1+u-s} = \sum_{u=0}^{s-1} \bar{\beta}_u h_{\mu+1+u-s} = h_{\mu+1}$$

where the second equation is because of the induction assumption for $\ell \leq \mu$, $\hat{h}_\ell = h_\ell$.

Thus, we have $h_\ell = \hat{h}_\ell$ for all ℓ, which means $\mathbf{h} = \hat{\mathbf{h}} = \mathbf{C}^\dagger \gamma$. Thus $\mathbf{t}$, the IDFT of $\mathbf{h}$, becomes $\mathbf{t} = \mathbf{Ch} = \mathbf{CC}^\dagger \gamma = \gamma$. Then the returned Index Set $V = \{j : e_{j+1} \neq 0\} = \{j : \gamma_j \neq 0\}$. By Lemma A.2, with probability 1, $\{j : \gamma_j \neq 0\} = \{j : \|\mathbf{n}_j\|_0 \neq 0\}$. Therefore, we have with probability 1, $V = \{j : \|\mathbf{n}_j\|_0 \neq 0\}$, which finishes the proof. □

A.5. Proof of Theorem 3.5

We first prove the correctness of the cyclic code. By Lemma 3.4, the set U contains the index of all non-adversarial compute nodes with probability 1. By Lemma 3.3, there exists $\mathbf{b}$ such that $\mathbf{W}_{\cdot,U} b = \mathbf{1}$. Therefore, $\mathbf{u}^{Cyc} = \mathbf{R}^{Cyc}_{\cdot,U}\mathbf{b} = (\mathbf{GW} + \mathbf{N})_{\cdot,U}\mathbf{b} = \mathbf{GW}_{\cdot,U}\mathbf{b} = \mathbf{G1}_P$. Thus, The cyclic code $(\mathbf{A}^{Cyc}, E^{Cyc}, D^{Cyc})$ can recover the desired gradient and hence resist any $\leq s$ adversaries with probability 1.

Next we show the efficiency of the cyclic code. By the construction of $\mathbf{A}^{Cyc}$ and $\mathbf{W}$, the redundancy ratio is $2s + 1$ which reaches the lower bound. Each compute node needs to compute a linear combination of the gradients of the data it holds, which needs $\mathcal{O}((2s + 1)d)$ computations. For the PS, the detection function $\phi(\cdot)$ takes $\mathcal{O}(d)$ (generating the random vector $\mathbf{f}$) + $\mathcal{O}(dP + 2Ps)$ (computing $\mathbf{fRC}_R^\dagger$) + $\mathcal{O}(s^2)$ (solving the Toeplitz system of linear equations in (A.1)) + $\mathcal{O}((P - 2s)s)$ (computing $h_\ell, \ell = 0, 1, 2, \cdots, P - 2s - 1$) + $\mathcal{O}(P \log P)$ (computing the DFT of $\mathbf{h}$) + $\mathcal{O}(P)$ (examining the non-zero elements of $\mathbf{t}$) = $\mathcal{O}(d + dP + 2Ps + s^2 + (P - 2s)s + P \log P + P) = \mathcal{O}(dP + Ps + P \log P)$. Finding the vector $\mathbf{b}$ takes $\mathcal{O}(P^3)$ (by simply constructing $\mathbf{b}$ via $[\mathbf{C}_L]_{\cdot,\bar{U}}$, though better algorithms may exist). The recovering equation $\mathbf{R}_{\cdot,U}\mathbf{b}$ takes $O(dP)$. Thus, in total, the decoder at the PS takes $\mathcal{O}(dP + P^3 + P \log P)$. When $d \gg P$, *i.e.*, $d = \Omega(P^2)$, this becomes $\mathcal{O}(dP)$. Therefore, $(\mathbf{A}^{Cyc}, E^{Cyc}, D^{Cyc})$ also achieves linear-time encoding and decoding. □

B. Streaming Majority Vote Algorithm

In this section we present the Boyer-Moore majority vote algorithm (Boyer & Moore, 1991), which is an algorithm that only needs computation linear in the size of the sequence.

Algorithm 2 Streaming Majority Vote.

```
Input  : $n$ items $I_1, I_2, \cdots, I_n$
Output : The majority of the $n$ items
Initialize an element $Ma = I_1$ and a counter $Counter = 0$.
for $i = 1$ to $n$ do
    if $Counter == 0$ then
        $Ma = I_i$.
        $Counter = 1$.
    else if $Ma == I_i$ then
        $Counter = Counter + 1$.
    else
        $Counter = Counter - 1$.
    end
end
Return $Ma$.
```

Clearly this algorithm runs in linear time and it is known that if there is a majority item then the algorithm finally will return it (Boyer & Moore, 1991).

C. Additional Experimental Results

C.1. End-to-end Convergence Performance

Table 4. Speedups (*i*.e., X times faster) of DRACO (Repetition/Cyclic Codes) over GM when using a fully-connected neural network on the MNIST dataset. We run both methods until they reach the same specified testing accuracy. In the table 'const' and 'rev grad' refer to the two types of adversarial updates.

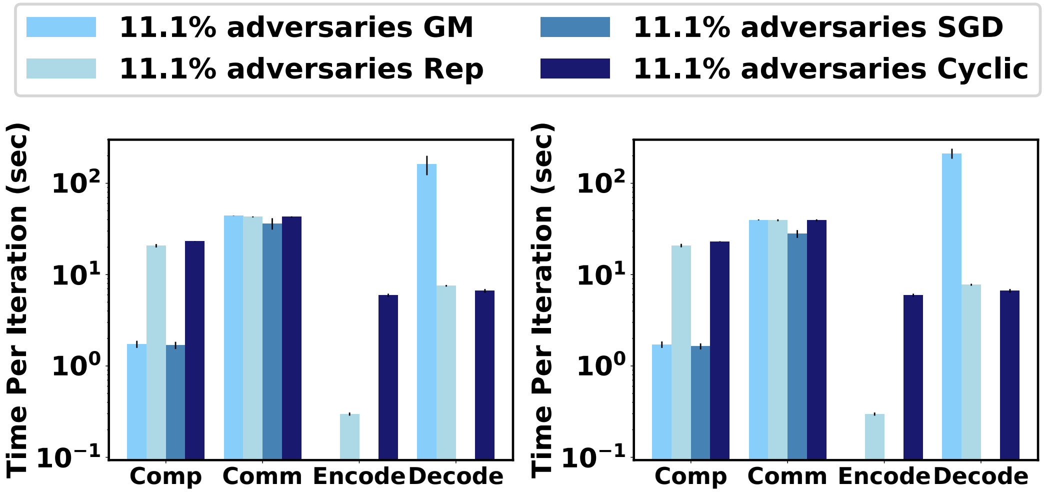
Test Accuracy	80%	85%	88%	90%
2.2% const	**3.4/2.7**	**3.5/2.8**	**4.8/3.9**	**4.1/3.1**
6.7% const	**2.7/2.0**	**4.1/3.1**	**6.0/4.6**	**5.6/4.1**
11.1% const	**2.9/2.2**	**4.8/3.7**	**6.1/4.7**	**5.3/3.8**
2.2% rev grad	**2.2/1.9**	**2.4/2.2**	**4.1/3.7**	**3.2/2.9**
6.7% rev grad	**3.1/2.5**	**3.3/3.1**	**5.5/4.8**	**4.5/3.7**
11.1% rev grad	**2.7/2.3**	**3.0/2.6**	**3.1/2.7**	**3.1/2.6**

C.2. Runtime Analysis for Large Model

We provide the large model runtime analysis for ResNet-152 here. As shown in Figure 5, we observe a trend similar to that from VGG and AlexNet. The decoding time of the GM approach is significantly higher than that of DRACO. DRACO, as expected, is several times faster than the GM approach in terms of the total runtime.

C.3. Effects of number of adversaries

We also analyze how the number of adversaries affects the performance of DRACO. We ran Cifar10 on ResNet-18 with 15 compute nodes, varying the number of adversaries s from 1 to 7. For these experiments, we used the constant adversary model. For the repetition code, we adapted the group size based on s while in the cyclic code we always took $2s + 1$. Figure 6 shows the total runtime cost of DRACO does not increase significantly as the number of adversaries increase. This is likely due to the fact that even at $s = 7$, the communication cost (which is not affected by the number of stragglers) is the dominant cost of the algorithm.



(a) Rev Grad

(b) Const

Figure 5. Empirical Per Iteration Time Cost on ResNet-152 with 11.1% adversarial nodes (a): reverse gradient adversary, (b): constant adversary

Table 5. Averaged Per Iteration Time Costs on ResNet-152 with 11.1% adversary

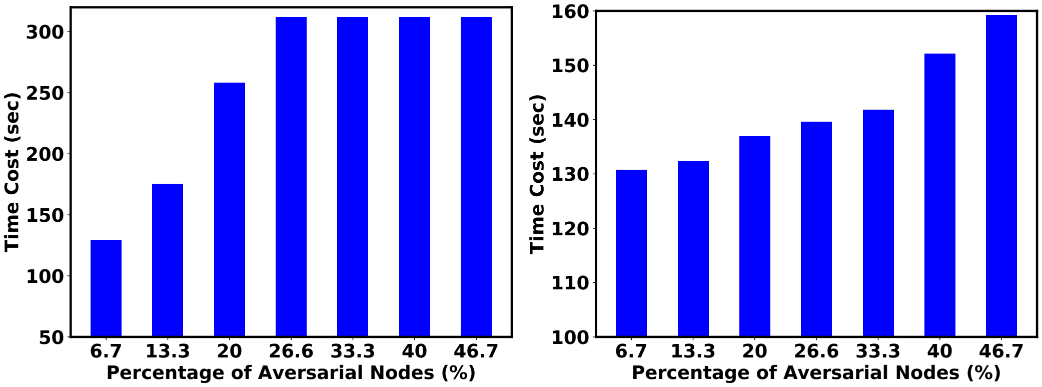
Time Cost (sec)	Comp	Comm	Encode	Decode
GM const	1.72	39.74	0	212.31
Rep const	20.81	39.36	0.24	7.74
SGD const	1.64	27.99	0	0.09
Cyclic const	23.08	39.36	5.94	6.64
GM rev grad	1.73	43.98	0	161.29
Rep rev grad	20.71	42.86	0.29	7.54
SGD rev grad	1.69	36.27	0	0.09
Cyclic rev grad	23.08	42.86	5.95	6.65

Table 6. Averaged Per Iteration Time Costs on VGG-19 with 11.1% adversary

Time Cost (sec)	Comp	Comm	Encode	Decode
GM const	0.26	12.47	0	74.63
Rep const	2.59	12.91	0.20	3.03
SGD const	0.25	6.9	0	0.03
Cyclic const	3.08	12.91	4.01	4.30
GM rev grad	0.26	14.57	0	39.02
Rep rev grad	2.55	14.66	0.20	3.04
SGD rev grad	0.25	7.15	0	0.03
Cyclic rev grad	3.07	14.66	4.02	3.65

Table 7. Averaged Per Iteration Time Costs on AlexNet with 11.1% adversarial nodes.

Time Cost (sec)	Comp	Comm	Encode	Decode
GM const	0.37	27.40	0	275.08
Rep const	4.16	30.71	0.67	10.65
SGD const	0.35	25.72	0	0.14
Cyclic const	3.67	30.71	13.55	12.54
GM rev grad	0.36	28.10	0	163.48
Rep rev grad	4.15	31.76	0.67	9.98
SGD rev grad	0.35	26.76	0	0.11
Cyclic rev grad	3.66	31.755	13.55	12.54



(a) Repetition Code

(b) Cyclic Code

Figure 6. Time Cost to Reach 70% Test set Accuracy with Cifar10 dataset run with ResNet-18 on cluster 15 computation nodes varying Percentage of Adversarial Nodes from 6.7% to 46.7% with Constant Adversary (a) Repetition Code and (b) Cyclic Code