Combinatorial Quantitative Group Testing with Adversarially Perturbed Measurements

Yun-Han Li and I-Hsiang Wang
Graduate Institute of Communication Engineering,
National Taiwan University, Taipei, Taiwan
Email: {r07942058,ihwang}@ntu.edu.tw

***Abstract*—In this paper, combinatorial quantitative group testing (QGT) with noisy measurements is studied. The goal of QGT is to detect defective items from a data set of size n with counting measurements, each of which counts the number of defects in a selected pool of items. While most literatures consider either probabilistic QGT with random noise or combinatorial QGT with noiseless measurements, our focus is on the combinatorial QGT with noisy measurements that might be adversarially perturbed by additive bounded noises. Since perfect detection is impossible, a partial detection criterion is adopted. With the adversarial noise being bounded by $d_n = \Theta(n^\delta)$ and the detection criterion being to ensure no more than $k_n = \Theta(n^\kappa)$ errors can be made, our goal is to characterize the fundamental limit on the number of measurement, termed pooling complexity, as well as provide explicit construction of measurement plans with optimal pooling complexity and efficient decoding algorithms. We first show that the fundamental limit is $\frac{1}{1-2\delta}\frac{n}{\log n}$ to within a constant factor not depending on (n, κ, δ) for the non-adaptive setting when $0 < 2\delta \le \kappa < 1$, sharpening the previous result by Chen and Wang [1]. We also provide deterministic constructions of an adaptive method with $\frac{1}{1-2\delta}\frac{n}{\log_2 n}$ pooling complexity up to a constant factor and $O(n)$ decoding complexity. While explicit construction of optimal non-adaptive methods remains open, a reduction to a simple combinatorial problem is proposed.**

An extended version of this paper is accessible at:
http://homepage.ntu.edu.tw/~ihwang/Eprint/itw20cqgt.pdf

I. Introduction

Group testing is the problem of identifying defective items in a large set with cardinality n by taking measurements on pools (subsets) of items. The type of measurement plays a central role in the fundamental limits of detection efficiency. In a classical model by Dorfman [2], binary-valued measurements are considered, where the output is a bit indicating the existence of defected items in the measured pool. Extensive results for this model (termed traditional group testing hereafter), including algorithms and information theoretic limits, can be found in surveys [3], [4] and the references therein.

Meanwhile, in many modern applications such as bioinformatics [5], network traffic monitoring [6], resource allocation in multi-user communication systems [7], etc., more informative measurement on the pool of items can be carried out. A natural one is the counting measurement that outputs the number of defective items in the pool. This is called the *quantitative group testing* (QGT) problem or the coin weighing problem with its root in combinatorics dating back to Shapiro [8]. QGT with noiseless measurements has been extensively studied. In particular, it has been shown that the minimum number of measurements is asymptotically $\frac{2n}{\log_2 n}$ [9] with explicit constructions of the optimal non-adaptive measurement plans [10], [11]. These results are combinatorial in nature as the goal is to detect the defects no matter where they are located. Hence, it is also called the combinatorial QGT (CQGT) with noiseless measurements, to contrast another more recent line of works that takes a probabilistic approach [12], [13], termed probabilistic QGT hereafter.

In practice, however, measurement might be noisy, as counting the number of defectives might be too costly to be accurate. In database applications, in order to preserve privacy, the measurement might also be perturbed intentionally. While the traditional group testing with noisy measurements has been extensively studied (see [4] for a survey), QGT with noisy measurements is far less understood. One line of works pertains to probabilistic QGT with *random* perturbation in the measurement [14]. Another line of works [1], [15]–[17] consider CQGT with adversarially perturbed measurements. It has been shown in [1] that, for $\delta, \kappa \in (0, 1)$, when the perturbation is at most the order of $\Theta(n^\delta)$ and the goal is to detect the defective items within Hamming distance at most the order of $\Theta(n^\kappa)$, there is a sharp phase transition in the fundamental limit: for $0 < 2\delta \le \kappa < 1$, the optimal pooling complexity is $\Theta(\frac{n}{\log n})$, and for $0 < \kappa < 2\delta < 1$, it is $\omega(n^p)\ \forall\, p \in \mathbb{N}$. This sharpened results in previous works related to data privacy [15], [16]. However, unlike the noiseless case [10], [11], only the existence of good measurement plans was shown in [1] by a probabilistic argument, and the optimal explicit construction remained open.

In this work, we improve the previous work [1] both in characterization of the optimal pooling complexity and construction of algorithms for the regime $0 < 2\delta \le \kappa < 1$. As for the information theoretic limit, we characterize the relationship between (κ, δ) and the leading coefficient of the optimal non-adaptive pooling complexity which turns out to be $\frac{1}{1-2\delta}\frac{n}{\log n}$ to within a constant factor not depending on (κ, δ). We further investigate the sparse CQGT (SCQGT) problem, that is, the original CQGT problem with an additional condition that the number of defective items is not greater than a threshold that we term the *sparsity* level. When the sparsity level is $\Theta(n^\lambda)$, for $0 < 2\delta \le \kappa < \lambda < 1$, the optimal pooling complexity is also characterized to within a constant factor not depending on $(\kappa, \delta, \lambda)$. Achievability is proved via a probabilistic argument, and the converse proof extends that of Erdös and Rényi [9].

As for the construction of CQGT algorithms, the following contributions are made. We first provide an explicit construction of a non-adaptive measurement plan with pooling complexity being $\frac{1}{\kappa-2\delta}\frac{n}{\log_2 n}$ (which is not optimal in the leading coefficient) to within a constant factor, along with a procedure that combines this construction with any sufficiently good SCQGT algorithms to reach a construction that has the optimal pooling complexity. The whole problem boils down to the design of good SCQGT algorithms. We further provide an explicit adaptive SCQGT algorithm that meet the above-mentioned criterion, leading to an overall adaptive measurement plan with pooling complexity being $\frac{1}{1-2\delta}\frac{n}{\log_2 n}$ to within a constant factor, together with $O(n)$ decoding complexity. As for the construction of the non-adaptive SCQGT algorithm, we reduce it to a simple combinatorial problem, and the existence of the solution is proved by a probabilistic method. As a result, if one can provide an explicit construction for this simple combinatorial problem, we can construct a SCQGT non-adaptive measurement plan with close-to-optimal pooling complexity. Consequently, the construction of an optimal non-adaptive measurement plan for CQGT can be completed.

Related Works

There are several closely related works [17]–[20] that consider CQGT with adversarially perturbed measurements. The noise model in the measurement, however, are all quite different from ours. In [17], there are three possible outcomes: the correct sum, an erroneous outcome with an arbitrary value, and an erasure symbol "?". When the total number of erroneous (or erasure) outcomes is assumed to be at most a fraction of the total number of measurements, which can be viewed as a ℓ_0-*norm* constraint on the perturbation vector, the optimal non-adaptive pooling complexity is characterized to within a constant factor. Another line of related works pertain to the binary multiple-access adder channel [18]–[20], where the perturbation vector is constrained in the ℓ_1-*norm*. In contrast, the noise model in our work constrains the perturbation vector in the ℓ_∞-*norm*, which makes perfect detection impossible, while in the related works mentioned above, only the perfect-detection criterion is considered.

II. Problem Formulation

In this section, let us define the combinatorial quantitative group testing (CQGT) problem and other related notions. A CQGT problem comprises the following:

Data: for each item indexed by $j = 1, ..., n$, we use $x_j \in \{0, 1\}$ to denote whether or not the j-th item is defective. Hence, the n-by-1 *data vector* $\boldsymbol{x} := [x_1\ x_2\ ...\ x_n]^\intercal$ is the target to be reconstructed from the noisy measurements.

Counting measurements: the pool of items in the i-th counting measurement can be represented by an 1-by-n *pooling vector* $\boldsymbol{q_i} \in \{0, 1\}^{1\times n}$, and the outcome of the counting measurement is $\boldsymbol{q}_i\boldsymbol{x}$. For a non-adaptive pooling algorithm, the measurement plan can be concisely represented by an n-by-s *pooling matrix* $\mathbf{Q}$ with its i-th row being the i-th pooling vector $\boldsymbol{q}_i$. Here s denotes the number of measurements, termed *pooling complexity*.

Perturbed outcomes: the outcome of the i-th measurement is $y_i = \boldsymbol{q}_i\boldsymbol{x} + n_i$, where $n_i \in [-d_n, d_n]$ denotes the bounded additive perturbation in the i-th measurement. The s outcomes of the measurements can be written as an s-by-1 vector $\boldsymbol{y} = \mathbf{Q}\boldsymbol{x}+\boldsymbol{n}$, where $\boldsymbol{n}$ is the *perturbation vector* with $\|\boldsymbol{n}\|_\infty \le d_n$.

Detection: for any data vector $\boldsymbol{x} \in \{0, 1\}^{n\times 1}$, the estimate generated by the detection algorithm (denoted by $\hat{\boldsymbol{x}}$) should be close to $\boldsymbol{x}$. In particular, the Hamming distance between $\hat{\boldsymbol{x}}$ and $\boldsymbol{x}$ should not be greater than k_n, that is, $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_1 \le k_n$.

Hence, a pooling matrix $\mathbf{Q}$ that solves the above CQGT problem if and only if

$$\begin{aligned} &\forall\, \boldsymbol{x}, \boldsymbol{x}' \in \{0, 1\}^{n\times 1} \text{ with } \|\boldsymbol{x} - \boldsymbol{x}'\|_1 > k_n, \\ &\|\mathbf{Q}\boldsymbol{x} - \mathbf{Q}\boldsymbol{x}'\|_\infty > d_n. \end{aligned} \tag{1}$$

Let us introduce the following definition.

Definition 2.1: (n, k_n, d_n)-CQGT denotes the combinatorial quantitative group testing problem defined above. If a pooling matrix $\mathbf{Q}$ is a solution to (n, k_n, d_n)-CQGT, it is called an (n, k_n, d_n)-detecting matrix. $s^*_{\mathsf{CQGT}}(n, k_n, d_n)$ denotes the smallest possible pooling complexity among all non-adaptive pooling algorithms, that is, it is the smallest height of (n, k_n, d_n)-detecting matrices.

Throughout our development, it turns out that CQGT with an additional sparsity constraint, which we call *sparse combinatorial group testing* (SCQGT), can be explored simultaneously. The optimal pooling complexity for non-adaptive SCQGT is also characterized, along with efficient adaptive algorithms. Let us introduce the following definition to better refer to this problem.

Definition 2.2: (n, k_n, d_n, l_n)-SCQGT denotes the combinatorial quantitative group testing problem (n, k_n, d_n)-CQGT with the additional sparsity assumption on the data vector $\boldsymbol{x}$, that is, $\|\boldsymbol{x}\|_1 \le l_n$. If a pooling matrix $\mathbf{Q}$ is a solution to (n, k_n, d_n, l_n)-SCQGT, with a slight abuse of notation, it is called an (n, k_n, d_n, l_n)-detecting matrix. $s^*_{\mathsf{SCQGT}}(n, k_n, d_n, l_n)$ denotes the smallest possible pooling complexity among all non-adaptive pooling algorithms, that is, it is the smallest height of (n, k_n, d_n, l_n)-detecting matrices.

III. Fundamental Limits

In this section, we provide the characterization of the optimal non-adaptive pooling complexity for (n, n^κ, n^δ)-CQGT, $0 < 2\delta \le \kappa < 1$. The characterization is tight to within a constant factor that is independent of (n, κ, δ), as stated in the following theorem.

Theorem 3.1: For $0 < 2\delta \le \kappa < 1$,

$$s^*_{\mathsf{CQGT}}(n, n^\kappa, n^\delta) = \frac{1}{1-2\delta}\frac{n}{\log n}$$

up to a constant factor that is independent of (n, κ, δ).

Proof: The proof comprises two parts: achievability and converse, established in the lemmas below.

Lemma 3.1 (CQGT Achievability): For $0 < 2\delta \le \kappa < 1$,

$$\limsup_{n\to\infty} \frac{s^*_{\mathsf{CQGT}}(n, n^\kappa, n^\delta)}{n/\log n} \le \frac{8}{1-2\delta}\ .$$

In words, there exists a sequence of (n, n^κ, n^δ)-detecting matrices with pooling complexity not greater than $\frac{8}{1-2\delta}\frac{n}{\log n}$ as $n \to \infty$.

Lemma 3.2 (CQGT Converse): For $0 < 2\delta \le \kappa < 1$,

$$s^*_{\mathsf{CQGT}}(n, n^\kappa, n^\delta) \ge \tfrac{2}{1-2\delta}\tfrac{n}{\log n}.$$

The two lemmas complete the proof of the theorem. The proof of achievability (Lemma 3.1) is in Appendix A of the full version, which uses a probabilistic argument to prove the existence of good pooling matrices. Converse (Lemma 3.2) is proved in Appendix B of the full version, which is based on extending a counting argument with its root in [9]. ∎

It is interesting to note that the leading coefficient does not depend on the order of the detection criterion κ. In other words, as long as partial detection to within n^κ successfully detection items is allowed, the number of pools to be measured only depend on the strength of the adversarial perturbation n^δ, where $\delta \le \kappa/2$.

When the defective items are sparsely populated in the data set, the number of pools needed to be measured should be smaller. The following theorem characterizes the optimal non-adaptive pooling complexity for $(n, n^\kappa, n^\delta, n^\lambda)$-SCQGT when $0 < 2\delta \le \kappa < \lambda < 1$.

Theorem 3.2: For $0 < 2\delta \le \kappa < \lambda < 1$,

$$s^*_{\mathsf{SCQGT}}(n, n^\kappa, n^\delta, n^\lambda) = \begin{cases} \frac{1-\lambda}{\lambda-2\delta} n^\lambda, & 2\delta < \kappa \\ \frac{1-\lambda}{\lambda-2\delta} n^\lambda \log n, & 2\delta = \kappa \end{cases}$$

up to a constant factor that is independent of $(n, \kappa, \delta, \lambda)$.

Proof: Similar to the proof of Theorem 3.1, the following two lemmas correspond to achievability and converse respectively, and their combination completes the proof.

Lemma 3.3 (SCQGT Achievability): For $0 < 2\delta \le \kappa < \lambda < 1$,

$$\limsup_{n\to\infty} \frac{s^*_{\mathsf{SCQGT}}(n,n^\kappa,n^\delta,n^\lambda)}{n^\lambda} \le \tfrac{4(1-\lambda)}{\lambda-2\delta}, \qquad 2\delta < \kappa$$
$$\limsup_{n\to\infty} \frac{s^*_{\mathsf{SCQGT}}(n,n^\kappa,n^\delta,n^\lambda)}{n^\lambda \log n} \le \tfrac{4(1-\lambda)}{\lambda-2\delta}, \qquad 2\delta = \kappa$$

Lemma 3.4 (SCQGT Converse): For $0 < 2\delta \le \kappa < \lambda < 1$,

$$s^*_{\mathsf{SCQGT}}(n, n^\kappa, n^\delta, n^\lambda) \ge \begin{cases} \frac{2(1-\lambda)}{\lambda-2\delta} n^\lambda, & 2\delta < \kappa \\ \frac{2(1-\lambda)}{\lambda-2\delta} n^\lambda \log n, & 2\delta = \kappa \end{cases}$$

Proofs of the above two lemmas are similar to those of Lemma 3.1 and 3.2 and hence left in the appendices of the extended version. ∎

IV. Algorithms

In this section, first we give a basic construction of a non-adaptive measurement plan for (n, n^κ, n^δ)-CQGT with pooling complexity that has the optimal order in n but a suboptimal leading coefficient in terms of κ, δ in Section IV-A. To achieve a better leading coefficient, an adaptive pooling algorithm for SCQGT is developed in Section IV-B. This adaptive pooling algorithm is then combined with the basic non-adaptive measurement plan in Section IV-A to give an explicit adaptive pooling algorithm with a pooling complexity that has a matching leading coefficient with the optimal non-adaptive one. While the explicit construction of a good *non-adaptive* measurement plan for $(n, n^\kappa, n^\delta, n^\lambda)$-SCQGT is still missing, we provide reduction of this problem to a simpler combinatorial problem detailed in Section IV-C. This non-adaptive measurement plan would then be combined with the basic non-adaptive measurement plan to give an optimal *non-adaptive* pooling algorithm.

A. Basic construction

The basic construction of the non-adaptive CQGT measurement plan is given below. Some necessary notations are set up first. Let $\epsilon = \kappa - 2\delta > 0$. Let $\|\lceil n^{\epsilon/2}\rceil\|$ denote the smallest possible width of the detecting matrix for the noiseless coin weighing problem mentioned in Section 4 of [11] that is not smaller than $n^{\epsilon/2}$, and $\mathbf{M}_{\|\lceil n^{\epsilon/2}\rceil\|}$ be the corresponding detecting matrix. Let $\|\|\lceil n^{1-\epsilon/2}\rceil\|\|$ denote the smallest possible size of the Sylvester's type Hadamard matrix that is not smaller than $n^{1-\epsilon/2}$, and $\mathbf{H}_{\|\|\lceil n^{1-\epsilon/2}\rceil\|\|}$ be the corresponding Hadamard matrix. Let $\bar{n} = \|\lceil n^{\epsilon/2}\rceil\| \, \|\|\lceil n^{1-\epsilon/2}\rceil\|\|$ and let

$$\mathbf{P}_{\bar{n}} = \mathbf{M}_{\|\lceil n^{\epsilon/2}\rceil\|} \otimes \mathbf{H}_{\|\|\lceil n^{1-\epsilon/2}\rceil\|\|},$$

the Kronecker product of the two matrices.

We are ready to give our basic construction. According to the above setup, entries of $\mathbf{P}_{\bar{n}}$ take value in $\{0, \pm 1\}$. Let us find two $\{0,1\}$-matrices $\mathbf{Q}^1_{\bar{n}}$ and $\mathbf{Q}^2_{\bar{n}}$ such that $\mathbf{Q}^1_{\bar{n}} - \mathbf{Q}^2_{\bar{n}} = \mathbf{P}_{\bar{n}}$, concatenate them vertically into a new matrix $\mathbf{Q}$, and delete the last $\bar{n} - n$ columns of $\mathbf{Q}$ to get $\hat{\mathbf{Q}}$. The width of the matrix $\hat{\mathbf{Q}}$ becomes n, and $\hat{\mathbf{Q}}$ stands for the measurement matrix that we would like to construct.

The basic construction $\hat{\mathbf{Q}}$ turns out to be a detecting matrix for CQGT with guarantees summarized in following theorem, the proof of which is detailed in Appendix C of the full version. It leverages the structure of the Hadamard matrix along with the detecting capability of $\mathbf{M}$.

Theorem 4.1 (Basic Construction): For n sufficiently large, $\hat{\mathbf{Q}}$ is a $\left(n, n^\kappa, n^\delta\right)$-detecting matrix with pooling complexity no more than $\frac{48}{\kappa-2\delta}\frac{n}{\log_2(n)}$.

In Appendix D of the full version, we also provide a companion two-step decoding algorithm for this non-adaptive measurement plan with time complexity $O(n)$.

B. Explicit construction of an adaptive pooling algorithm

Towards improving the leading coefficient of pooling complexity, we first give an adaptive pooling algorithm for SCQGT. The key ingredient of this algorithm is divide-and-conquer. The details of the algorithm are given step-by-step in the following. Let S be the index set of defect items.

0) Initialize $I \leftarrow \{1, ..., n\}$, the whole index set of items.
1) Divide I into $2n^\lambda$ equal-size segments (subsets) $I_1, I_2, ..., I_{2n^\lambda}$, and define $C = \{1, 2, ..., 2n^\lambda\}$, the index set of the segments. Let $k_i = |S \bigcap I_i|$, the number of detectives in I_i. Make $2n^\lambda$ measurements as follows. First prepare a Hadamard matrix $\mathbf{H}_{|C|^+}$ of size $|C|^+$, where $|C|^+$ denotes the smallest power of 2 not smaller than $|C|$, and then delete the last $|C|^+ - |C|$ columns

of $\mathbf{H}_{|C|^+}$. Denote this new matrix by $\mathbf{H}_{|C|}$. Replace $(\mathbf{H}_{|C|})_{i,j}$ by $(\mathbf{H}_{|C|})_{i,j}\mathbf{1}_{|I_j|}$, where $(\mathbf{H}_{|C|})_{i,j}$ denotes the i,j-th entry of $\mathbf{H}_{|C|}$ and $\mathbf{1}_{|I_j|}$ denotes the all-1 row vector with size $1 \times |I_j|$. The resulting new matrix $\hat{\mathbf{H}}$ serves as the pooling matrix that determines the $2n^\lambda$ counting measurements in this step.

2) After making the counting measurements in Step 1), we get outcome
$$\boldsymbol{y} = \hat{\mathbf{H}}\boldsymbol{x} + \boldsymbol{n} = \mathbf{H}_{|C|}[k_1\ k_2\ ...\ k_{|C|}]^\intercal + \boldsymbol{n}.$$
Let $[\hat{k}_1\ k_2\ ...\ \hat{k}_{|C|}]^\intercal = \lceil \mathbf{H}_{|C|}^{-1}\boldsymbol{y} \rfloor$, where $\lceil\rfloor$ denotes rounding to the closest integer.

3) Update C: for $i = 1, ..., |C|$, remove i from C if $\hat{k}_i = 0$.

4) Update $I \leftarrow \bigcup_{i\in C} I_i$. If $|I| > 2n^\lambda$, go back to Step 1). Otherwise, go to Step 5) to terminate.

5) Divide I into $|I|$ segments and prepare a Hadamard matrix $\mathbf{H}_{|I|^+}$ of size $|I|^+$, where $|I|^+$ is the smallest power of 2 not smaller than $|I|$. Delete the last $|I|^+ - |I|$ columns of $\mathbf{H}_{|I|^+}$ and denote this new matrix by $\mathbf{H}_{|I|}$, which is used for counting measurements to detect these I bits. After making these counting measurements, we get the outcome $\boldsymbol{y} = \mathbf{H}_{|I|}\boldsymbol{x} + \boldsymbol{n}$ and let $[\hat{k}_1, ..., \hat{k}_{|I|}]^\intercal = \lceil \mathbf{H}_{|I|}^{-1}\boldsymbol{y} \rfloor$. Finally, remove i from I if $\hat{k}_i = 0$, $\forall\, i$. Return the resulting I as the set of indices of the defective items.

Let us now analyze the performance of this adaptive pooling algorithm for SCQGT. The sparsity level is assumed to be n^λ. First, let us deal with the number of iterations. In Step 3), since the sparsity level is n^λ, there exists at least n^λ segments I_i such that k_i equals to 0. Hence $|I|$ is reduced at least by half in Step 3) of each iteration. As a result, after no more than $\log_2 n$ iterations, this algorithm enters Step 5) for termination.

As for the quality of detection, let $\boldsymbol{w} = [\hat{k}_1\ ...\ \hat{k}_{|C|}]^\intercal - [k_1\ ...\ k_{|C|}]^\intercal$ in each iteration. Since $\mathbf{H}_{|C|}$ has the property that $\forall \boldsymbol{x} \in \mathbb{R}^{|C|}$, $\|\mathbf{H}_{|C|}\boldsymbol{x}\|_2^2 = |C|^+ \|\boldsymbol{x}\|_2^2$, if $\|\boldsymbol{w}\|_2^2 > n^{2\delta}$, $\|\mathbf{H}_{|C|}\boldsymbol{w}\|_\infty \geq (\frac{\|\mathbf{H}_{|C|}\boldsymbol{w}\|_2^2}{|C|^+})^{1/2} = \|\boldsymbol{w}\|_2 > n^\delta$, contradicting our assumption about noise. Hence, $\|\boldsymbol{w}\|_2^2 \leq n^{2\delta}$, implying $\|\boldsymbol{w}\|_1 \leq n^{2\delta}$ since $\boldsymbol{w}$ is a $\{0, \pm 1\}$-vector. In each iteration, we throw away at most $n^{2\delta}$ indices that has non-zero value, and we iterate at most $\log_2 n$ times. In Step 5), we throw away at most $n^{2\delta}$ indices that has non-zero value too. Hence, we make at most $n^{2\delta}\log_2 n$ mistakes.

As for the pooling complexity, in each iteration, we make $2n^\lambda$ counting measurements, and in Step 5), we make at most $2n^\lambda$ counting measurements. Hence the total pooling complexity is $2n^\lambda \log_2 n$. It is also easy to check that total decoding complexity is $O(n^\lambda \log_2 n)$.

With the above discussions, we come up with the following theorem about the performance of this adaptive scheme.

Theorem 4.2 (A Sparse Pooling Algorithm): For SCQGT with n items and sparsity level n^λ, noise level n^δ, the proposed adaptive pooling algorithm uses no more than $2n^\lambda \log_2 n$ counting measurements, and it cause at most $n^{2\delta}\log_2 n$ mistakes with $O(n^\lambda \log_2 n)$ decoding complexity.

We are now ready to combine the non-adaptive measurement plan in Section IV-A and the aforementioned adaptive SCQGT pooling algorithm to produce an adaptive scheme for (n, n^κ, n^δ)-CQGT with pooling complexity $\frac{1}{1-2\delta}\frac{n}{\log_2 n}$ to within a constant factor. The overall procedure goes as follows.

Step A: First, employ the measurement plan of the basic construction in Section IV-A for (n, n^λ, n^δ)-CQGT and use the corresponding decoding algorithm in Appendix D. The decoded result is then represented as follows:
$$\hat{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{p} + \boldsymbol{q}, \tag{2}$$
where $\boldsymbol{x}$ is the true data vector and $\boldsymbol{p}, \boldsymbol{q}$ are n^λ-sparse $\{0,1\}$-vectors with non-overlapping supports. In other words, $I_{\boldsymbol{p}} \cap I_{\boldsymbol{q}} = \emptyset$ where $I_{\boldsymbol{p}}$ and $I_{\boldsymbol{q}}$ denote supports, that is, the index sets of non-zero elements, of $\boldsymbol{p}$ and $\boldsymbol{q}$ respectively. Note that we intentionally split the mistakes made in this initial decoded result into two parts $\boldsymbol{p}$ and $\boldsymbol{q}$, and they are treated separately in Step B and Step C below. Since $\hat{\boldsymbol{x}} - \boldsymbol{x}$ is a ternary vector taking values in $\{0, \pm 1\}$, binary vectors $\boldsymbol{p}, \boldsymbol{q}$ have unique solutions and therefore they are well-defined. In this step, we make $s_{\text{A}} = \frac{48}{\lambda - 2\delta}\frac{n}{\log_2 n}$ counting measurements.

Step B: In this step, $\boldsymbol{p}$, a first part of the mistakes made in Step A, will be detected. From (2), the support of $\boldsymbol{p}$ is contained in the complement of the support of $\hat{\boldsymbol{x}}$, that is, $I_{\boldsymbol{p}} \subseteq \overline{I}_{\hat{\boldsymbol{x}}} \triangleq \{1, ..., n\} \setminus I_{\hat{\boldsymbol{x}}}$. The aforementioned adaptive SCQGT pooling algorithm is then employed onto $\hat{\boldsymbol{x}}$ to detect the support of $\boldsymbol{p}$, viewing $\boldsymbol{x} + \boldsymbol{q}$ as adversarial perturbation. As a result, $\boldsymbol{p}$ can be decoded with at most $n^{2\delta}\log_2 n$ mistakes using no more than $s_{\text{B}} = 2n^\lambda \log_2 n$ counting measurements.

Step C: In this step, $\boldsymbol{q}$, the remaining part of the mistakes made in Step A, will be detected. From (2), $I_{\boldsymbol{q}} \subseteq I_{\hat{\boldsymbol{x}}}$. Once again, the adaptive SCQGT pooling algorithm is employed onto $\hat{\boldsymbol{x}}$ to detect the support of $\boldsymbol{q}$, viewing $\boldsymbol{x} + \boldsymbol{p}$ as adversarial perturbation. A slight twist is needed: in the original adaptive pooling algorithm, the number of 1's is counted, while here instead, the number of 0's is counted. As a result, $\boldsymbol{q}$ can be decoded with at most $n^{2\delta}\log_2 n$ mistakes using no more than $s_{\text{C}} 2n^\lambda \log_2 n$ counting measurements.

To sum up, the total number of mistakes made at the end is at most $2n^{2\delta}\log_2 n$ with the number of counting measurements no more than
$$s_{\text{total}} = s_{\text{A}} + s_{\text{B}} + s_{\text{C}} = \frac{48}{\lambda - 2\delta}\frac{n}{\log_2 n} + 4n^\lambda \log_2 n.$$
Asymptotically, when n is large enough, $s_{\text{total}} \approx \frac{48}{\lambda - 2\delta}\frac{n}{\log_2 n}$. Taking $\lambda \to 1$, s_{total} tends to $\frac{48}{1-2\delta}\frac{n}{\log_2 n}$. As a result, the total pooling complexity is $\frac{48}{1-2\delta}\frac{n}{\log_2 n}$. Moreover, and it is easy to check that the total decoding complexity is $O(n)$.

We summarize the above discussion about the performance of the proposed adaptive CQGT pooling algorithm that combine the basic non-adaptive scheme in Section IV-A in the theorem below.

Theorem 4.3: This deterministic adaptive pooling algorithm for (n, n^κ, n^δ)-CQGT has pooling complexity $\frac{48}{1-2\delta}\frac{n}{\log_2 n}$ and decoding complexity $O(n)$.

C. Towards constructing an optimal non-adaptive scheme

The scheme developed in Section IV-B is adaptive with pooling complexity matching the non-adaptive optimum.

Meanwhile, explicit construction of the optimal non-adaptive measurement plan is still missing, and a key bottleneck is the lack of sufficiently good explicit non-adaptive SCQGT pooling algorithms. In this section, we describe how to construct a leading-coefficient-optimal non-adaptive CQGT measurement plan if a sufficiently good explicit non-adaptive SCQGT scheme exists. Then, we reduce the problem of constructing non-adaptive SCQGT schemes to a simple combinatorial problem whose solution is shown to exist.

Suppose a sufficiently good non-adaptive measurement plan for $(n, n^\kappa, n^\delta, n^\lambda)$-SCQGT exists and its pooling matrix is $\mathbf{Q}$. Take the basic construction for (n, n^λ, n^δ)-CQGT in Section IV-A with pooling matrix being $\mathbf{M}$. Concatenate $\mathbf{M}$ and $\mathbf{Q}$ vertically to get $\mathbf{R}$ with $\mathbf{R}^\intercal = \begin{bmatrix}\mathbf{M}^\intercal & \mathbf{Q}^\intercal\end{bmatrix}$. By definition,

$$\|\mathbf{M}\boldsymbol{x}\|_\infty \geq n^\delta \quad \forall\, \boldsymbol{x} \in \{0, \pm 1\}^n \text{ with } \|\boldsymbol{x}\|_0 \geq n^\lambda,$$
$$\|\mathbf{Q}\boldsymbol{x}\|_\infty \geq n^\delta \quad \forall\, \boldsymbol{x} \in \{0, \pm 1\}^n \text{ with } 2n^\lambda \geq \|\boldsymbol{x}\|_0 \geq n^\kappa.$$

Hence, $\|\mathbf{R}\boldsymbol{x}\|_\infty = \max\{\|\mathbf{M}\boldsymbol{x}\|_\infty, \|\mathbf{Q}\boldsymbol{x}\|_\infty\} \geq n^\delta$ for all $\boldsymbol{x} \in \{0, \pm 1\}^n$ with $\|\boldsymbol{x}\|_0 \geq n^\kappa$. This suggests that $\mathbf{R}$, the vertical concatenation of $\mathbf{M}$ and $\mathbf{Q}$, is a detecting matrix for (n, n^κ, n^δ)-CQGT.

Its pooling complexity is the sum of those of $\mathbf{Q}$ and $\mathbf{M}$. Since the pooling complexity of $\mathbf{M}$ is $\frac{48}{\lambda - 2\delta}\frac{n}{\log_2 n}$, as long as that of $\mathbf{Q}$ is n^λ to within a poly-log factor, the overall pooling complexity can be made $\frac{48}{1-2\delta}\frac{n}{\log_2 n}$ similarly as in Section IV-B by setting $\lambda \to 1$.

With the above discussion, the remaining problem is how to construct $\mathbf{Q}$ with the desirable pooling complexity. To construct such a matrix, we propose an approach to reduce the original problem into a combinatorial problem, so that once the solution to the combinatorial problem is found, we are able to construct a $(n, n^\kappa, \frac{n^\delta}{48(\log n)^{3/2}}, n^\lambda)$-detecting matrix with height $192n^\lambda(\log n)^3$, satisfying the desirable property of $\mathbf{Q}$ mentioned above.

The reduction relies on a special form of the pooling matrix $\mathbf{Q}$ that we propose. In particular, we construct

$$\mathbf{Q} = \begin{bmatrix}\mathbf{D}_1^\intercal & \dots & \mathbf{D}_{4\log n}^\intercal\end{bmatrix}^\intercal, \tag{3}$$

where $\mathbf{D}_i, ..., \mathbf{D}_{4\log n}$ are designed as follows. Suppose there exists a binary matrix $\mathbf{B}$ with size $4n^\lambda \log(n) \times n$ that satisfies the following properties:

1) Each of its column vector is $4\log n$-sparse.
2) For any d of its column vectors $\boldsymbol{c}_{i_1}, ..., \boldsymbol{c}_{i_d}$, $1 \leq d \leq n^\lambda$, $\|\bigcup_{j=i_1,...,i_d} \boldsymbol{c}_j\|_0 \geq \frac{1}{6}d$, where $\bigcup$ denotes the bit-wise "or" operation of binary column vectors.

Finding such matrix $\mathbf{B}$ is a combinatorial problem, the existence of which is proved in Appendix F of the full version.

Let us now decompose this matrix into $4\log n$ binary matrices $\mathbf{B}_1, ..., \mathbf{B}_{4\log(n)}$ with each of their column has exactly one 1 and $\sum_{i=1}^{4\log n}\mathbf{B}_i = \mathbf{B}$. Then we introduce the check matrix for BCH code as our building block because it can perfectly distinguish those binary vectors with ℓ_0-norm smaller than a certain threshold (its code distance). Let $\mathbf{C}_t^n$ be the check matrix of a BCH code with size n and code distance t. Suppose there are n_j columns in $\mathbf{B}_i$ such that they have 1 at the j-th place, we replace these columns by $\mathbf{H}_{4n^\lambda \log n}(j) \otimes \mathbf{C}_{48\log n}^{n_j}$, where $\mathbf{H}_{4n^\lambda \log n}(j)$ denote the j-th column of the Hadamard matrix $\mathbf{H}_{4n^\lambda \log n}$ and $\otimes$ denotes the Kronecker product. The resulting matrices are our designed $\mathbf{D}_i$'s in (3). The following theorem summarizes the guarantees of the constructed $\mathbf{Q}$, and its proof can be found in Appendix E of the extended version.

Theorem 4.4: The constructed matrix $\mathbf{Q}$ in equation (3) is $(n, n^\kappa, \frac{n^\delta}{48(\log n)^{3/2}}, n^\lambda)$-detecting and its pooling complexity is at most $192n^\lambda(\log n)^3$.

REFERENCES

[1] W.-N. Chen and I.-H. Wang, "Partial data extraction via noisy histogram queries: Information theoretic bounds," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 2488–2492.

[2] R. Dorfman, "The detection of defective members of large populations," *The Annals of Mathematical Statistics*, vol. 14, no. 4, pp. 436–440, 1943.

[3] D. Du and F. Hwang, *Combinatorial group testing and its applications*. World Scientific, 1993.

[4] M. Aldridge, O. Johnson, and J. Scarlett, "Group testing: An information theory perspective," *Foundations and Trends® in Communications and Information Theory*, vol. 15, no. 3-4, pp. 196–392, 2019. [Online]. Available: http://dx.doi.org/10.1561/0100000099

[5] C.-C. Cao, C. Li, and X. Sun, "Quantitative group testing-based overlapping pool sequencing to identify rare variant carriers," *BMC Bioinformatics*, vol. 15, no. 1, p. 195, June 2014.

[6] C. Wang, Q. Zhao, and C. Chuah, "Group testing under sum observations for heavy hitter detection," in *2015 Information Theory and Applications Workshop (ITA)*, Feb 2015, pp. 149–153.

[7] G. De Marco, T. Jurdziński, and D. R. Kowalski, "Optimal channel utilization with limited feedback," in *Fundamentals of Computation Theory*, L. A. Gąsieniec, J. Jansson, and C. Levcopoulos, Eds. Cham: Springer International Publishing, 2019, pp. 140–152.

[8] H. S. Shapiro and N. J. Fine, "Problem e 1399," *The American Mathematical Monthly*, vol. 67, no. 7, pp. 697–698, 1960.

[9] P. Erdős and A. Rényi, "On two problems of information theory," 1963.

[10] B. Lindström, "On a combinatorial problem in number theory," *Canadian Mathematical Bulletin*, pp. 477–490, 1965.

[11] D. G. Cantor and W. H. Mills, "Determination of a subset from certain combinatorial properties," *Canadian Journal of Mathematics*, vol. 18, pp. 42–48, 1966.

[12] A. E. Alaoui, A. Ramdas, F. Krzakala, L. Zdeborová, and M. I. Jordan, "Decoding from pooled data: Sharp information-theoretic bounds," *SIAM Journal on Mathematics of Data Science*, vol. 1, no. 1, pp. 161–168, 2019.

[13] E. Karimi, F. Kazemi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson, "Sparse graph codes for non-adaptive quantitative group testing," *Proceedings of IEEE Information Theory Workshop*, 2019.

[14] J. Scarlett and V. Cevher, "Phase transitions in the pooled data problem," *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pp. 377–385, 2017.

[15] I. Dinur and K. Nissim, "Revealing information while preserving privacy," 2003.

[16] C. Dwork and S. Yekhanin, "New efficient attacks on statistical disclosure control mechanisms," in *Advances in Cryptology""CRYPTO 2008*, vol. 5157, 2008, pp. 469–480.

[17] N. H. Bshouty, "On the coin weighing problem with the presence of noise," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer Berlin Heidelberg, 2012, pp. 471–482.

[18] S.-C. Chang and E. J. Weldon, "Coding for t-user multiple-access channels," *IEEE Transactions on Information Theory*, vol. 25, no. 6, pp. 684–691, November 1979.

[19] J. H. Wilson, "Error-correcting codes for a t-user binary adder channel," *IEEE Transactions on Information Theory*, vol. 34, no. 4, pp. 888–890, July 1988.

[20] J. Cheng, K. Kamoi, and Y. Watanabe, "User identification by signature code for noisy multiple-access adder channel," *Proceedings of IEEE International Symposium on Information Theory*, pp. 1974–1977, 2006.

APPENDIX

A. Proof of Achievability (Lemma 3.1 and 3.3)

A probabilistic argument is used to show the existence of good pooling matrices. In particular, we are going to *upper bound* the probability that a randomly generated matrix with height s is *not* an (n, k_n, d_n)-detecting matrix. If this probability is strictly bounded below 1, then the existence of (n, k_n, d_n)-detecting matrices is established. In words, we are going to show that $s \leq \frac{8}{1-2\delta}\frac{n}{\log n}$ is a sufficient condition for the upper bound mentioned above being strictly less than 1.

Let us now describe the random pooling matrix ensemble employed in this probabilistic argument. To simplify the analysis, we focus on pooling matrices with $\{\pm 1\}$-entries. Note that any pooling vector with $\{\pm 1\}$-entries can be generated by taking the difference of two pooling vectors with $\{0, 1\}$-entries. Hence, at the end of our analysis, to conform with the original CQGT problem formulation, we need to double the pooling complexity upper bound. The random pooling matrix ensemble is generated as follows: each element of the matrix is drawn from $\{\pm 1\}$ uniformly at random, i.i.d. across all entries. With a slight abuse of notation, let $\mathbf{Q}$ denote this random matrix, that is,

$$(\mathbf{Q})_{i,j} \overset{\text{i.i.d.}}{\sim} \text{Unif}(\{\pm 1\}),\ \forall\, (i,j) \in \{1, ..., s\} \times \{1, ..., n\},$$

and let $\boldsymbol{Q}_i$ denote the i-th row of $\mathbf{Q}$.

Consider the event $\mathcal{E}$ that $\mathbf{Q}$ is not an (n, k_n, d_n)-detecting matrix. By definition (Definition 2.1),

$$\mathcal{E} = \left\{ \begin{array}{ll} \exists\, \boldsymbol{x}, \boldsymbol{x}' \in \{0,1\}^{n\times 1} \text{ with } & \|\boldsymbol{x} - \boldsymbol{x}'\|_1 > k_n \text{ and} \\ & \|\mathbf{Q}\boldsymbol{x} - \mathbf{Q}\boldsymbol{x}'\|_\infty \leq d_n \end{array} \right\}. \tag{4}$$

For notational convenience, let us introduce

$$D_a^b = \left\{ \boldsymbol{x} - \boldsymbol{y} \,\middle|\, \boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^{n\times 1}, a < \|\boldsymbol{x} - \boldsymbol{y}\|_1 \leq b \right\} \tag{5}$$

to denote the set of difference vectors of ℓ_1-norm ranging from a to b. With the notations above, the event $\mathcal{E}$ can be succinctly written as

$$\mathcal{E} = \bigcup_{\boldsymbol{d}\in D_{k_n}^n} \left\{ \|\mathbf{Q}\boldsymbol{d}\|_\infty \leq d_n \right\}.$$

Hence, by the Union Bound,

$$\begin{aligned} \Pr\{\mathcal{E}\} &\leq \sum_{\boldsymbol{d}\in D_{k_n}^n} \Pr\left\{ \|\mathbf{Q}\boldsymbol{d}\|_\infty \leq d_n \right\} \\ &= \sum_{\boldsymbol{d}\in D_{k_n}^n} \prod_{i=1}^{s} \Pr\left\{ \boldsymbol{Q}_i\boldsymbol{d} \leq d_n \right\} \end{aligned} \tag{6}$$

Noting that the event $\{\boldsymbol{Q}_i\boldsymbol{d} \leq d_n\}$ is equivalent to the event that out of $\|\boldsymbol{d}\|_1$ i.i.d. $\text{Unif}(\{\pm 1\})$ random variables, the number of $+1$ and the number of -1 differ by at most d_n, we have

$$\begin{aligned} &\Pr\left\{ \boldsymbol{Q}_i\boldsymbol{d} \leq d_n \right\} \\ &= \sum_{\ell:\|\boldsymbol{d}\|_1 - d_n \leq 2\ell \leq \|\boldsymbol{d}\|_1 + d_n} \binom{\|\boldsymbol{d}\|_1}{\ell} 2^{-\|\boldsymbol{d}\|_1} \\ &\leq d_n \binom{\|\boldsymbol{d}\|_1}{\lfloor \frac{1}{2}\|\boldsymbol{d}\|_1 \rfloor} 2^{-\|\boldsymbol{d}\|_1} \overset{\text{(a)}}{\leq} d_n \left(\pi \frac{\|\boldsymbol{d}\|_1}{2} \right)^{-1/2} \end{aligned} \tag{7}$$

(a) is due to the fact that $\binom{j}{\lfloor j/2 \rfloor} \leq 2^j \left(\pi j/2\right)^{-1/2}$ for all $j \in \mathbb{N}$. Combining (6) and (7), we get

$$\begin{aligned} \Pr\{\mathcal{E}\} &\leq \sum_{\boldsymbol{d}\in D_{k_n}^n} \left(d_n\sqrt{2/\pi} \right)^s \|\boldsymbol{d}\|_1^{-s/2} \\ &= \sum_{\ell=\lfloor k_n \rfloor + 1}^{n} |D_{\ell-1}^{\ell}| \left(d_n\sqrt{2/\pi} \right)^s \ell^{-s/2} \end{aligned} \tag{8}$$

To proceed, the range of the above summation is divided into three regimes and bounded separately: the first regime is $k_n \le \ell \le k_n n^{2\epsilon}$, the second regime is $k_n n^{2\epsilon} \le \ell \le n^{1-2\epsilon}$, and the third regime is $n^{1-2\epsilon} \le \ell \le n$, where ϵ is a positive constant that is smaller than $(1-\log_n k_n)/4 = (1-\kappa)/4$. Then,

$$\begin{aligned}
(8) &\overset{(a)}{\le} \left|D^{k_n n^{2\epsilon}}_{\lfloor k_n \rfloor}\right| \left(\frac{d_n}{\sqrt{k_n}}\sqrt{2/\pi}\right)^s \\
&\quad + \left|D^{n^{1-2\epsilon}}_{k_n n^{2\epsilon}}\right| \left(\frac{d_n}{\sqrt{k_n n^{2\epsilon}}}\sqrt{2/\pi}\right)^s \\
&\quad + \left|D^{n}_{n^{1-2\epsilon}}\right| \left(\frac{d_n}{\sqrt{n^{1-2\epsilon}}}\sqrt{2/\pi}\right)^s \\
&\overset{(b)}{\le} (2(n+1))^{k_n n^{2\epsilon}} \left(\frac{d_n}{\sqrt{k_n}}\sqrt{2/\pi}\right)^s \\
&\quad + (2(n+1))^{n^{1-2\epsilon}} \left(\frac{d_n}{\sqrt{k_n n^{2\epsilon}}}\sqrt{2/\pi}\right)^s \\
&\quad + 3^n \left(\frac{d_n}{\sqrt{n^{1-2\epsilon}}}\sqrt{2/\pi}\right)^s \\
&\overset{(c)}{=} (2(n+1))^{n^{\kappa+2\epsilon}} \left(n^{\delta-\frac{\kappa}{2}}\sqrt{2/\pi}\right)^s \qquad (9) \\
&\quad + (2(n+1))^{n^{1-2\epsilon}} \left(n^{\delta-\frac{\kappa}{2}-\epsilon}\sqrt{2/\pi}\right)^s \qquad (10) \\
&\quad + 3^n \left(n^{\delta-\frac{1}{2}+\epsilon}\sqrt{2/\pi}\right)^s. \qquad (11)
\end{aligned}$$

(a) follows from dividing the whole summation into the three regimes mentioned above and applying the trivial lower bound of ℓ in each regime. (b) follows from applying two different upper bounds on the sizes of difference sets:

$$\begin{aligned}
|D^b_a| &= \textstyle\sum_{j=a+1}^{b} \binom{n}{j} 2^j \le \sum_{j=0}^{b} \binom{n}{j} 2^b \le (n+1)^b 2^b, \qquad (12)\\
|D^b_a| &= \textstyle\sum_{j=a+1}^{b} \binom{n}{j} 2^j \le \sum_{j=0}^{n} \binom{n}{j} 2^j = 3^n.
\end{aligned}$$

(c) follows from plugging in $k_n = n^\kappa, d_n = n^\delta$.

Finally, in order to ensure all the three terms (9) – (11) vanish as $n \to \infty$, since it is the most stringent to drive (11) to zero, it suffices to choose

$$s = \frac{\log 3}{1/2 - \delta - \epsilon}\frac{n}{\log n}.$$

Picking sufficiently small $\epsilon \in (0, \frac{1-\kappa}{4})$ such that it is smaller than $(\frac{1}{2}-\delta)(1-\frac{\log 3}{2})$, we immediately see that it is also sufficient to choose

$$s = \frac{4}{1-2\delta}\frac{n}{\log n}$$

to ensure (9) – (11) all vanish as $n \to \infty$. As a result, there exists a $\{\pm 1\}$-pooling matrix with size $s = \frac{4}{1-2\delta}\frac{n}{\log n}$. Finally, note that a $\{0,1\}$-pooling matrix can be generated by simple row operations from $\{\pm 1\}$-pooling matrix, with the increase of the height by at most a factor of 2. Hence, there exists a binary pooling matrix with size $s = \frac{8}{1-2\delta}\frac{n}{\log n}$, and this completes the proof of Lemma 3.1.

As for the proof of achievability for the sparse case (Lemma 3.3), we slightly modify the definition of event $\mathcal{E}$ in (4) as follows:

$$\mathcal{E} = \left\{ \begin{array}{ll} \exists\, \boldsymbol{x}, \boldsymbol{x}' \in \{0,1\}^{n\times 1} \text{ with } & \|\boldsymbol{x}\|_1, \|\boldsymbol{x}'\|_1 < l_n \\ & \|\boldsymbol{x}-\boldsymbol{x}'\|_1 > k_n \text{ and} \\ & \|\mathbf{Q}\boldsymbol{x} - \mathbf{Q}\boldsymbol{x}'\|_\infty \le d_n \end{array} \right\}.$$

In words, it is the event that $\mathbf{Q}$ is not an (n, k_n, d_n, l_n)-detecting matrix. Then, following the same proof program, an upper bound on the probability of this event, similar to (8), can be found as follows:

$$\Pr\{\mathcal{E}\} \le \sum_{\ell=\lfloor k_n \rfloor + 1}^{2l_n} |\tilde{D}^\ell_{\ell-1}| \left(d_n\sqrt{2/\pi}\right)^s \ell^{-s/2}. \qquad (13)$$

Note that now in the definition of the difference set $\tilde{D}^b_a$, there is an additional condition $\|\boldsymbol{x}\|, \|\boldsymbol{y}\| \le l_n$, compared to that of D^b_a in (5). Next, following the steps to upper bound (8) by (9) – (11), we derive an upper bound on (13) by diving the range

of the above summation into three regimes and bounding them separately: the first regime is $k_n \le \ell \le k_n n^{2\epsilon}$, the second regime is $k_n n^{2\epsilon} \le \ell \le l_n n^{-2\epsilon}$, and the third regime is $l_n n^{-2\epsilon} \le \ell \le 2l_n$, where ϵ is a positive constant that is smaller than $(\log_n l_n - \log_n k_n)/4 = (\lambda - \kappa)/4$. Then,

$$\begin{aligned}
(13) &\le \left| D^{k_n n^{2\epsilon}}_{\lfloor k_n \rfloor} \right| \left(\frac{d_n}{\sqrt{k_n}} \sqrt{2/\pi} \right)^s \\
&\quad + \left| D^{l_n n^{-2\epsilon}}_{k_n n^{2\epsilon}} \right| \left(\frac{d_n}{\sqrt{k_n n^{2\epsilon}}} \sqrt{2/\pi} \right)^s \\
&\quad + \left| D^{2l_n}_{l_n n^{-2\epsilon}} \right| \left(\frac{d_n}{\sqrt{l_n n^{-2\epsilon}}} \sqrt{2/\pi} \right)^s \\
&\overset{\text{(d)}}{\le} (2(n+1))^{k_n n^{2\epsilon}} \left(\frac{d_n}{\sqrt{k_n}} \sqrt{2/\pi} \right)^s \\
&\quad + (2(n+1))^{l_n n^{-2\epsilon}} \left(\frac{d_n}{\sqrt{k_n n^{2\epsilon}}} \sqrt{2/\pi} \right)^s \\
&\quad + 2^{2l_n (\log_2(\frac{en}{2l_n}) + 1)} \left(\frac{d_n}{\sqrt{l_n n^{-2\epsilon}}} \sqrt{2/\pi} \right)^s \\
&\overset{\text{(e)}}{=} (2(n+1))^{n^{\kappa+2\epsilon}} \left(n^{\delta - \frac{\kappa}{2}} \sqrt{2/\pi} \right)^s \qquad (14) \\
&\quad + (2(n+1))^{n^{\lambda-2\epsilon}} \left(n^{\delta - \frac{\kappa}{2} - \epsilon} \sqrt{2/\pi} \right)^s \qquad (15) \\
&\quad + 2^{2n^{\lambda}((1-\lambda)\log_2(n) + \log_2(\mathrm{e}))} \left(n^{\delta - \frac{\lambda}{2} + \epsilon} \sqrt{2/\pi} \right)^s . \qquad (16)
\end{aligned}$$

(d) follows from (12), $|\tilde{D}^b_a| \le \sum_{j=0}^b \binom{n}{j} 2^b$, and

$$\textstyle\sum_{j=0}^b \binom{n}{j} \le \sum_{j=0}^b \frac{n^j}{j!} = \sum_{j=0}^b \frac{b^j}{j!} \left(\frac{n}{b}\right)^j \le \mathrm{e}^b \left(\frac{n}{b}\right)^b .$$

(e) follows from plugging in $l_n = n^\lambda, k_n = n^\kappa, d_n = n^\delta, 0 < 2\delta \le \kappa < \lambda < 1$.

Then, following the same discussion in the non-sparse case, in order to make (14) – (16) all vanish as $n \to \infty$, it suffices to choose a sufficiently small ϵ and

$$s = \begin{cases} \frac{2(1-\lambda)}{\lambda - 2\delta} n^\lambda, & 2\delta < \kappa \\ \frac{2(1-\lambda)}{\lambda - 2\delta} n^\lambda \log_2 n, & 2\delta = \kappa \end{cases}$$

As a result, there exists a $\{0,1\}$-pooling matrix with size

$$s = \begin{cases} \frac{4(1-\lambda)}{\lambda - 2\delta} n^\lambda, & 2\delta < \kappa \\ \frac{4(1-\lambda)}{\lambda - 2\delta} n^\lambda \log_2 n, & 2\delta = \kappa \end{cases}$$

B. Proof of Converse (Lemma 3.2 and 3.4)

The proof of converse is based on packing. It will be shown that if a pooling matrix $\mathbf{Q}$ is (n, k_n, d_n)-detecting, the number of measurements s (the height of $\mathbf{Q}$) must be greater than or equal to a certain threshold. The argument goes as follows. Note that the detection criterion (1) implies that, for any k_n-packing C_G with respect to ℓ_1-norm of a subset $G \subseteq \{0,1\}^n$, its image set after multiplying with $\mathbf{Q}$, $\mathbf{Q}[C_G] \triangleq \{\mathbf{Q}\boldsymbol{x} \,|\, \boldsymbol{x} \in C_G\}$, must be a d_n-packing with respect to ℓ_∞-norm of the image set $\mathbf{Q}[G] \triangleq \{\mathbf{Q}\boldsymbol{x} \,|\, \boldsymbol{x} \in G\}$. By properly choosing G, one can derive a good upper bound on the packing number of $\mathbf{Q}[G]$ which is related to s, the height of $\mathbf{Q}$. Meanwhile, a lower bound of the packing number of G is also a lower bound of the packing number of $\mathbf{Q}[G]$, which can be found by a simple counting argument. The two bounds are then combined to derive a lower bound of s.

Let us consider the k_n-packing number of G with respect to ℓ_1-norm. Note that it is lower bounded by the k_n-covering number of G with respect to ℓ_1-norm, and the covering number is further lowered by $|G|$ divided by the cardinality of an ℓ_1-norm ball with radius k_n. Hence, there exists a maximal packing C_G with

$$|C_G| \ge \frac{|G|}{\sum_{j=0}^{k_n} \binom{n}{j}} \ge \frac{|G|}{(n+1)^{k_n}}. \qquad (17)$$

The choice of the subset G is a second key to the proof. Since we are going to upper bound the d_n-packing number with respect to ℓ_∞-norm of the image set $\mathbf{Q}[G]$, we select G so that it is *strictly* contained in an s-dimensional cube of appropriate

side lengths, say, $r_1, r_2, ..., r_s$. Then, as $\boldsymbol{Q}_i\boldsymbol{x}$ are integers for all $\boldsymbol{x}$ and $i = 1, ..., s$, the packing number with respect to ℓ_∞-norm is upper bounded by

$$\prod_{i=1}^{s} \frac{r_i}{d_n} = \frac{\prod_{i=1}^{s} r_i}{d_n^s}. \tag{18}$$

Combining (17) and (18), it can be seen that with $d_n = n^\delta$ and $k_n = n^\kappa$,

$$\begin{aligned} &\frac{\prod_{i=1}^{s} r_i}{d_n^s} \geq \frac{|G|}{(n+1)^{k_n}} \implies \\ &s\left(\sum_{i=1}^{s} \frac{\log r_i}{s} - \delta \log n\right) \geq \log|G| - n^\kappa \log(n+1). \end{aligned} \tag{19}$$

Hence, to get the desired lower bound on s, $r_i \approx \sqrt{n}$ within a poly-logarithm factor and $|G| \approx 2^n$.

The above discussion motivates us to make the following choice of G. Let $q_i \triangleq \frac{1}{2}\left\|\boldsymbol{Q}_i\right\|_1$, the number of 1's in the i-th counting measurement, $i = 1, ..., s$. To this end, let us define "atypical" sets to be excluded as follows: for $i = 1, 2, ..., s$,

$$\begin{aligned} &\text{If } q_i \geq \sqrt{n \log n}, \\ &\quad B_i \triangleq \left\{\boldsymbol{x} \in \{0,1\}^n : |\boldsymbol{Q}_i\boldsymbol{x} - q_i/2| \geq \sqrt{q_i \log q_i}\right\}. \end{aligned} \tag{20}$$

$$\begin{aligned} &\text{If } q_i < \sqrt{n \log n}, \\ &\quad B_i \triangleq \emptyset. \end{aligned} \tag{21}$$

The "typical" set to be considered is hence

$$G \triangleq \{0,1\}^n \setminus B, \text{ where } B \triangleq \textstyle\bigcup_{i=1}^{s} B_i. \tag{22}$$

To control the cardinality of B_i, let us employ Hoeffiding's inequality as follows: randomize the data vector so that the n elements $X_1, ..., X_n$ are now n i.i.d. $\mathrm{Ber}(1/2)$ random variables. In other words, $\boldsymbol{X} = [X_1\ X_2\ ...X_n]^\intercal \sim \mathrm{Unif}(\{0,1\}^n)$. As a result, for (20), $|B_i| = 2^n \Pr\left\{|\boldsymbol{Q}_i\boldsymbol{X} - q_i/2| \geq r_i/2\right\}$, with $r_i = 2\sqrt{q_i \log q_i}$. Note that given a pooling vector $\boldsymbol{Q}_i$, the outcome of the counting measurement, $\boldsymbol{Q}_i\boldsymbol{X}$, is just the sum of q_i i.i.d. $\mathrm{Ber}(1/2)$ random variables. Hence, by Hoeffiding's inequality, with $r_i = 2\sqrt{q_i \log q_i}$,

$$\Pr\left\{|\boldsymbol{Q}_i\boldsymbol{X} - q_i/2| \geq r_i/2\right\} \leq 2\mathrm{e}^{-r_i^2/2q_i} = 2q_i^{-2} \leq \tfrac{2}{n \log n}.$$

Consequently, $\forall\, i = 1, ..., s$, $|B_i| \leq 2^n \frac{2}{n \log n}$, and

$$|G| \geq 2^n - \sum_{i=1}^{s} |B_i| \geq 2^n\Big(1 - \frac{2s}{n \log n}\Big) \geq 2^n\Big(1 - \frac{2}{\log n}\Big), \tag{23}$$

where in the last inequality, we make use of an implicit assumption that $n \geq s$ when n is sufficiently large, due to the achievability part (Lemma 3.1).

Let us now turn back to inequality (19). The choice of G in (20) – (22) together with the fact that $0 \leq \boldsymbol{Q}_i\boldsymbol{x} \leq q_i$ (since it is the outcome of a counting measurement with q_i items in the pool) ensures that the image set $\mathbf{Q}[G]$ is strictly contained in an s-dimensional cube with side lengths not greater than $2\sqrt{n \log n}$. Hence, (19) and (23) imply

$$\begin{aligned} &s\Big(\log(2\sqrt{n \log n}) - \delta \log n\Big) \\ &\geq \log\left(2^n\Big(1 - \frac{2}{\log n}\Big)\right) - n^\kappa \log(n+1). \end{aligned}$$

As n tends to infinity, we conclude that

$$\liminf_{n\to\infty} \frac{s}{n/\log n} \geq \frac{1}{1-2\delta}\ .$$

The proof for the sparse case (Lemma 3.4) largely follows that of the non-sparse case, with slight modification of the definition of the "atypical" sets in (20) and (21): for $i = 1, 2, ..., s$, the definition of B_i in (20) is changed to

$$\left\{\boldsymbol{x} \in \{0,1\}^n : \|\boldsymbol{x}\|_1 \leq n^\lambda, |\boldsymbol{Q}_i\boldsymbol{x} - \tfrac{q_i n^\lambda}{n}| \geq \sqrt{6\lambda n^\lambda \log n}\right\}.$$

Accordingly, the "typical" set G becomes

$$G \triangleq \{x \in \{0,1\}^n, \|\boldsymbol{x}\|_1 \leq n^\lambda\} \setminus B, \text{ where } B \triangleq \textstyle\bigcup_{i=1}^{s} B_i.$$

Chernoff bound is then employed to control the cardinality of B_i. Note that the new definition of B_i has an additional *sparsity* constraint $\|\boldsymbol{x}\|_1 \leq n^\lambda$. Removing the sparsity constraint, we have a set $\tilde{B}_i$ with cardinality not smaller than that of B_i. Now, randomize the data vector so that $X_i \overset{\text{i.i.d.}}{\sim} \text{Ber}(n^{\lambda-1})$, $i = 1, ..., n$. We first calculate $\Pr\left\{|\boldsymbol{Q}_i\boldsymbol{X} - q_i n^\lambda/n| \geq \sqrt{6\lambda n^\lambda \log n}\right\}$, and then relate this quantity to $|\tilde{B}_i|$.

$$
\begin{aligned}
&\Pr\left\{|\boldsymbol{Q}_i\boldsymbol{X} - \frac{q_i n^\lambda}{n}| \geq \sqrt{6\lambda n^\lambda \log n}\right\} \\
&\leq^{(a)} 2\frac{\left(n^{\lambda-1}e^t + 1 - n^{\lambda-1}\right)^{q_i}}{e^{t\left(q_i n^{\lambda-1} + \sqrt{6\lambda n^\lambda \log n}\right)}} = 2\frac{\left(1 + n^{\lambda-1}\left(e^t - 1\right)\right)^{q_i}}{e^{t\left(q_i n^{\lambda-1} + \sqrt{6\lambda n^\lambda \log n}\right)}} \\
&\leq 2\frac{e^{n^{\lambda-1}\left(e^t-1\right)q_i}}{e^{t\left(q_i n^{\lambda-1} + \sqrt{6\lambda n^\lambda \log n}\right)}} \leq^{(b)} \frac{2}{\left(n^\lambda \log n\right)^2}.
\end{aligned}
$$

(a) follows from Chernoff bound. In order to get (b), it suffics to choose t such that

$$
\begin{aligned}
&t\left(q_i n^{\lambda-1} + \sqrt{6\lambda n^\lambda \log n}\right) - n^{\lambda-1}\left(e^t - 1\right) q_i &(24)\\
&\geq 2\lambda \log(n) + 2\log\left(\log(n)\right) &(25)
\end{aligned}
$$

Then we have

$$
\begin{aligned}
&(24) \geq t\left(q_i n^{\lambda-1} + \sqrt{6\lambda n^\lambda \log n}\right) - n^{\lambda-1}\left(e^t - 1\right) q_i \\
&=^{(a)} \ln\left(1 + \frac{\sqrt{6\lambda n^\lambda \log n}}{n^{\lambda-1} q_i}\right)\left(q_i n^{\lambda-1} + \sqrt{6\lambda n^\lambda \log n}\right) \\
&- \sqrt{6\lambda n^\lambda \log n} := f(q_i) \\
&\geq^{(b)} \ln\left(1 + \frac{\sqrt{6\lambda n^\lambda \log n}}{n^\lambda}\right)\left(n^\lambda + \sqrt{6\lambda n^\lambda \log n}\right) \\
&- \sqrt{6\lambda n^\lambda \log n} \\
&\geq^{(c)} \left(\frac{\sqrt{6\lambda n^\lambda \log n}}{n^\lambda} - \frac{6\lambda n^\lambda \log n}{2n^{2\lambda}}\right) \\
&\left(n^\lambda + \sqrt{6\lambda n^\lambda \log n}\right) - \sqrt{6\lambda n^\lambda \log n} \\
&= \frac{1}{2}\frac{6\lambda n^\lambda \log n}{n^\lambda}\left(1 - \frac{\sqrt{6\lambda n^\lambda \log n}}{n^\lambda}\right) \\
&\geq 2\lambda \log n + 2\log\left(\log n\right)
\end{aligned}
$$

(a) follows from choose $t = \ln\left(1 + \frac{\sqrt{6\lambda n^\lambda \log n}}{n^\lambda}\right)$.$(b)$ follows from the the fact that $f(q_i)$ is a decreasing function of q_i and $1 \leq q_i \leq n$. (c) follows from $\ln(1+x) \geq x - \frac{x^2}{2}$.

Consequently, $\forall\, i = 1, ..., s$, $|B_i| \leq^{(a)} \binom{n}{n^\lambda} n^\lambda \frac{2}{n^{2\lambda}(\log n)^2}$, and

$$
\begin{aligned}
|G| &\geq \binom{n}{n^\lambda} - \sum_{i=1}^{s} |B_i| \geq \binom{n}{n^\lambda}\left(1 - \frac{2s}{n^\lambda (\log n)^2}\right) &(26)\\
&\geq^{(b)} n^{(1-\lambda)n^\lambda}\left(1 - \Theta(\frac{1}{\log n})\right), &(27)
\end{aligned}
$$

(a) follows from the fact that for all $x \in \{0,1\}^n, \|\boldsymbol{x}\|_1 \leq n^\lambda$, those $\|\boldsymbol{x}\|_1 = n^\lambda$ has the smallest probability $\left(\frac{n^\lambda}{n}\right)^{n^\lambda}\left(1 - \frac{n^\lambda}{n}\right)^{n-n^\lambda}$, and $\binom{n}{n^\lambda}\left(\frac{n^\lambda}{n}\right)^{n^\lambda}\left(1 - \frac{n^\lambda}{n}\right)^{n-n^\lambda} \geq \frac{1}{n^\lambda}$. where in (b), we make use of an implicit assumption that $s = O(n^\lambda \log n)$ when n is sufficiently large, due to the achievability part (Lemma 3.3), and the fact that $\binom{n}{n^\lambda} \geq n^{(1-\lambda)n^\lambda}$.

Finally, combine (19),(26), we get

$$s\Big(\log(2\sqrt{6\lambda n^{\lambda}\log n})-\delta\log n\Big)$$
$$\geq \log\left(n^{(1-\lambda)n^{\lambda}}\left(1-\Theta(\frac{1}{\log n})\right)\right)-n^{\kappa}\log(n+1).$$

As n tends to infinity, we conclude that

$$s\geq\begin{cases}\frac{2(1-\lambda)}{\lambda-2\delta}n^{\lambda}, & 2\delta<\kappa\\ \frac{2(1-\lambda)}{\lambda-2\delta}n^{\lambda}\log n, & 2\delta=\kappa\end{cases}$$

C. Proof of Theorem 4.1

Let $\epsilon\in(0,1)$, and $\lceil n^{\frac{\epsilon}{2}}\rceil$ denote the smallest width of noiseless detecting matrix corresponding to non-adaptive pooling algorithm mentioned in Section 4 of [11] that greater or equal than $n^{\frac{\epsilon}{2}}$. Let $\lceil n^{1-\frac{\epsilon}{2}}\rceil$ denote the smallest size of Sylvester's type Hadamard matrix that greater or equal than $n^{1-\frac{\epsilon}{2}}$. Let $\bar{n}=\lceil n^{1-\frac{\epsilon}{2}}\rceil\lceil n^{\frac{\epsilon}{2}}\rceil$. One can easily get that $n^{\frac{\epsilon}{2}}\leq\lceil n^{\frac{\epsilon}{2}}\rceil\leq 3n^{\frac{\epsilon}{2}}$, and $n^{1-\frac{\epsilon}{2}}\leq\lceil n^{1-\frac{\epsilon}{2}}\rceil\leq 2n^{1-\frac{\epsilon}{2}}$, and consequently, $n\leq\bar{n}\leq 6n$.

In order to proof that $\hat{\mathbf{Q}}$ is a $\left(n,n^{\kappa},n^{\delta}\right)$-detecting matrix, we need to show that $\forall\boldsymbol{a}\neq\boldsymbol{b}\in\{0,1\}^{n},\|\boldsymbol{a}-\boldsymbol{b}\|_{0}\geq n^{\kappa}$, $\left\|\hat{\mathbf{Q}}\left(\boldsymbol{a}-\boldsymbol{b}\right)\right\|_{\infty}\geq 2n^{\delta}$. Since $\hat{\mathbf{Q}}$ is reduced(delete last $\bar{n}-n$ column) from $\mathbf{Q}$, and $\mathbf{Q}$ is the concatenate of $\mathbf{Q}_{\bar{n}}^{1},\mathbf{Q}_{\bar{n}}^{2}$, and $\mathbf{P}_{\bar{n}}=\mathbf{Q}_{\bar{n}}^{1}-\mathbf{Q}_{\bar{n}}^{2}$. It suffics to show that for any $\boldsymbol{a}\neq\boldsymbol{b}\in\{0,1\}^{\bar{n}},\|\boldsymbol{a}-\boldsymbol{b}\|_{0}\geq n^{\kappa}$, $\left\|\mathbf{P}_{\bar{n}}\left(\boldsymbol{a}-\boldsymbol{b}\right)\right\|_{\infty}\geq 2n^{\delta}$.

Let $\boldsymbol{d}=\boldsymbol{a}-\boldsymbol{b}=[\boldsymbol{d}_{1},\boldsymbol{d}_{2},...,\boldsymbol{d}_{\lceil n^{1-\frac{\epsilon}{2}}\rceil}]$ be the equal length division of some difference vector, where $\boldsymbol{a}\neq\boldsymbol{b}\in\{0,1\}^{\bar{n}}$. Let $\boldsymbol{y}=[\boldsymbol{y}_{1},\boldsymbol{y}_{2},...,\boldsymbol{y}_{\lceil n^{1-\frac{\epsilon}{2}}\rceil}]=\mathbf{P}_{\bar{n}}\boldsymbol{d}$ be the equal length division of result vector. Since rows of Hadamard matrix form an orthogonal basis,

$$\|\boldsymbol{y}\|^{2}=\|\mathbf{P}_{\bar{n}}\boldsymbol{d}\|^{2}=\sum_{i=1}^{\lceil n^{1-\frac{\epsilon}{2}}\rceil}\lceil n^{1-\frac{\epsilon}{2}}\rceil\left\|\mathbf{M}_{\lceil n^{\frac{\epsilon}{2}}\rceil}\boldsymbol{d}_{i}\right\|^{2}$$

and $\mathbf{M}_{\lceil n^{\frac{\epsilon}{2}}\rceil}$ is a noiseless detecting matrix, so for any $\boldsymbol{d}_{i}\neq\mathbf{0}$, $\mathbf{M}_{\lceil n^{\frac{\epsilon}{2}}\rceil}\boldsymbol{d}_{i}\neq\mathbf{0}$, combine with the fact that $\mathbf{M}_{\lceil n^{\frac{\epsilon}{2}}\rceil}\boldsymbol{d}_{i}$ is an integer vector, $\left\|\mathbf{M}_{\lceil n^{\frac{\epsilon}{2}}\rceil}\boldsymbol{d}_{i}\right\|^{2}\geq 1$. but in our setting, $\|\boldsymbol{d}\|_{0}\geq n^{\kappa}$, so, there exists at least $\frac{n^{\kappa}}{\lceil n^{\frac{\epsilon}{2}}\rceil}$ segments $\boldsymbol{d}_{i}\neq\mathbf{0}$, hence

$$\begin{aligned}\|\boldsymbol{y}\|^{2}&=\sum_{i=1}^{\lceil n^{1-\frac{\epsilon}{2}}\rceil}\lceil n^{1-\frac{\epsilon}{2}}\rceil\left\|\mathbf{M}_{\lceil n^{\frac{\epsilon}{2}}\rceil}\boldsymbol{d}_{i}\right\|^{2}\\&\geq\lceil n^{1-\frac{\epsilon}{2}}\rceil\frac{n^{\kappa}}{\lceil n^{\frac{\epsilon}{2}}\rceil}\geq\frac{n^{1+\kappa-\epsilon}}{3}\end{aligned}$$

finally, remember that the height of $\boldsymbol{y}$ equals to the height of $\mathbf{M}_{\lceil n^{\frac{\epsilon}{2}}\rceil}$ times the height of $\mathbf{H}_{\lceil n^{1-\frac{\epsilon}{2}}\rceil}$, which is

$$\left(\frac{2\lceil n^{\frac{\epsilon}{2}}\rceil}{\log_{2}\left(\lceil n^{\frac{\epsilon}{2}}\rceil\right)}+O\left(\frac{\lceil n^{\frac{\epsilon}{2}}\rceil\log_{2}\left(\log_{2}\left(\lceil n^{\frac{\epsilon}{2}}\rceil\right)\right)}{\log_{2}^{2}\left(\lceil n^{\frac{\epsilon}{2}}\rceil\right)}\right)\right)\tag{28}$$

$$\lceil n^{1-\frac{\epsilon}{2}}\rceil\leq\frac{12n}{\log_{2}\left(n^{\frac{\epsilon}{2}}\right)}+O\left(\frac{n\log_{2}\left(\log_{2}\left(n\right)\right)}{\log_{2}^{2}\left(n\right)}\right)\tag{29}$$

$$\leq^{(1)}\frac{48n}{\epsilon\log_{2}\left(n\right)}=^{(2)}\frac{48n}{(\kappa-2\delta)\log_{2}\left(n\right)}=o\left(\frac{n}{3}\right)\tag{30}$$

When n large enough, inequality (1) follows. Equality (2) follows from that we choose $\epsilon=\kappa-2\delta$. So we have,

$$\|\boldsymbol{y}\|_{\infty}\geq\sqrt{\frac{4\|\boldsymbol{y}\|^{2}}{\frac{n}{3}}}\geq\sqrt{4n^{(1+\kappa-\epsilon)-1}}=\sqrt{4n^{\kappa-\epsilon}}=^{(1)}2n^{\delta}$$

Equality (1) follows from that we choose $\epsilon=\kappa-2\delta$. Hence we can see that, for any $\boldsymbol{a}\neq\boldsymbol{b}\in\{0,1\}^{\bar{n}},\|\boldsymbol{a}-\boldsymbol{b}\|_{0}\geq n^{\kappa}$, $\left\|\mathbf{P}_{\bar{n}}\left(\boldsymbol{a}-\boldsymbol{b}\right)\right\|_{\infty}\geq 2n^{\delta}$, which implies $\hat{\mathbf{Q}}$ is a $\left(n,n^{\kappa},n^{\delta}\right)$-detecting matrix, and by (30), the pooling complexity of this construction is no more than $\frac{48n}{(\kappa-2\delta)\log_{2}(n)}$.

D. Decode Algorithm for Basic Code

We propose a two step $O(n)$ decode algorithm :

1) Deconstruction step
2) Rounding step

1) Deconstruction Step: Let $\boldsymbol{y}^{'} = \hat{\mathbf{Q}}\boldsymbol{x}^{'} + \boldsymbol{n}^{'}, \boldsymbol{x}^{'} \in \{0,1\}^{n}$. Since $\hat{\mathbf{Q}}$ is reduced from $\mathbf{Q}$, $\boldsymbol{y}^{'} = \mathbf{Q}\boldsymbol{x} + \boldsymbol{n}^{'}, \boldsymbol{x} = \left[\boldsymbol{x}^{'}, \mathbf{0}_{\bar{n}-n}\right] \in \{0,1\}^{\bar{n}}$, where $\mathbf{0}_{\bar{n}-n}$ is the zero vector with size $\bar{n}-n$. Let $\epsilon = \kappa - 2\delta$. We first subtract $\boldsymbol{y}_u^{'}$(upper half $\boldsymbol{y}^{'}$) by $\boldsymbol{y}_l^{'}$(lower half of $\boldsymbol{y}^{'}$), then

$$\boldsymbol{y} = \boldsymbol{y}_u^{'} - \boldsymbol{y}_l^{'} = \left(\mathbf{Q}_n^1 - \mathbf{Q}_n^2\right)\boldsymbol{x} + \boldsymbol{n}_u^{'} - \boldsymbol{n}_l^{'} = \mathbf{P}_{\bar{n}}\boldsymbol{x} + \boldsymbol{n}$$

Note that Sylvester's type Hadamard matrix with size 2^d can be write as kronecker product of $\mathbf{H}_2$ itself d times

$$\mathbf{H}_{2^d} = \mathbf{H}_2 \otimes \mathbf{H}_2 \otimes ... \otimes \mathbf{H}_2 = \mathbf{H}_2^{\otimes d}$$

hence

$$\begin{aligned}\mathbf{P}_{\bar{n}} &= \mathbf{M}_{\left\lceil n^{\frac{\epsilon}{2}}\right\rceil} \otimes \mathbf{H}_{\left\lceil n^{1-\frac{\epsilon}{2}}\right\rceil} = \mathbf{M}_{\left\lceil n^{\frac{\epsilon}{2}}\right\rceil} \otimes \mathbf{H}_{\frac{\left\lceil n^{1-\frac{\epsilon}{2}}\right\rceil}{2}} \otimes \mathbf{H}_2 \\ &= \mathbf{P}_{\frac{\bar{n}}{2}} \otimes \mathbf{H}_2 = \begin{pmatrix} \mathbf{P}_{\frac{\bar{n}}{2}} & \mathbf{P}_{\frac{\bar{n}}{2}} \\ \mathbf{P}_{\frac{\bar{n}}{2}} & -\mathbf{P}_{\frac{\bar{n}}{2}} \end{pmatrix}\end{aligned}$$

then we can see that

$$\boldsymbol{y} = \begin{pmatrix} \boldsymbol{y}_u \\ \boldsymbol{y}_l \end{pmatrix} = \begin{pmatrix} \mathbf{P}_{\frac{\bar{n}}{2}} & \mathbf{P}_{\frac{\bar{n}}{2}} \\ \mathbf{P}_{\frac{\bar{n}}{2}} & -\mathbf{P}_{\frac{\bar{n}}{2}} \end{pmatrix}\boldsymbol{x} + \boldsymbol{n}$$

Next we do some row operation(deconstruction)

$$\begin{pmatrix} \frac{\boldsymbol{y}_u+\boldsymbol{y}_l}{2} \\ \frac{\boldsymbol{y}_u-\boldsymbol{y}_l}{2} \end{pmatrix} = \begin{pmatrix} \mathbf{P}_{\frac{\bar{n}}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\frac{\bar{n}}{2}} \end{pmatrix}\boldsymbol{x} + \begin{pmatrix} \frac{\boldsymbol{n}_u+\boldsymbol{n}_l}{2} \\ \frac{\boldsymbol{n}_u-\boldsymbol{n}_l}{2} \end{pmatrix}$$

After some calculation

$$\begin{aligned}\left\|\begin{pmatrix} \frac{\boldsymbol{n}_u+\boldsymbol{n}_l}{2} \\ \frac{\boldsymbol{n}_u-\boldsymbol{n}_l}{2} \end{pmatrix}\right\|^2 &= \left\|\frac{\boldsymbol{n}_u+\boldsymbol{n}_l}{2}\right\|^2 + \left\|\frac{\boldsymbol{n}_u-\boldsymbol{n}_l}{2}\right\|^2 \\ &= \frac{\|\boldsymbol{n}_u\|^2 + \|\boldsymbol{n}_l\|^2}{2} = \frac{\|\boldsymbol{n}\|^2}{2}\end{aligned}$$

We can see that the two-norm square of noise vector reduce by half after one time deconstruction. Hence, after we do $\log_2\left(\left\lceil n^{1-\frac{\epsilon}{2}}\right\rceil\right)$ times deconstruction

$$R\left(\boldsymbol{y}, \log_2\left(\left\lceil n^{1-\frac{\epsilon}{2}}\right\rceil\right)\right) = \tag{31}$$

$$\begin{pmatrix} \mathbf{P}_{\left\lceil n^{\frac{\epsilon}{2}}\right\rceil} & \mathbf{0} & ... & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\left\lceil n^{\frac{\epsilon}{2}}\right\rceil} & ... & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & ... & \mathbf{P}_{\left\lceil n^{\frac{\epsilon}{2}}\right\rceil} \end{pmatrix} \begin{pmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \vdots \\ \boldsymbol{x}_{\left\lceil n^{1-\frac{\epsilon}{2}}\right\rceil} \end{pmatrix} \tag{32}$$

$$+ R\left(\boldsymbol{n}, \log_2\left(\left\lceil n^{1-\frac{\epsilon}{2}}\right\rceil\right)\right) \tag{33}$$

Where we define $R(\boldsymbol{y}, t), R(\boldsymbol{n}, t)$ be the corresponding vector after doing t times deconstruction on column vector $\boldsymbol{y}, \boldsymbol{n}$. Since $\|\boldsymbol{n}\|_\infty \leq n^\delta$, and $s = o(n)$

$$\left\|R\left(\boldsymbol{n}, \log_2\left(\left\lceil n^{1-\frac{\epsilon}{2}}\right\rceil\right)\right)\right\|^2 = \frac{\|\boldsymbol{n}\|^2}{\left\lceil n^{1-\frac{\epsilon}{2}}\right\rceil} = \frac{o\left(n^{1+2\delta}\right)}{\left\lceil n^{1-\frac{\epsilon}{2}}\right\rceil} \tag{34}$$

$$= o\left(n^{2\delta+\frac{\epsilon}{2}}\right) \tag{35}$$

and then we divide $R\left(\boldsymbol{y}, \log_2\left(\left\lceil n^{1-\frac{\epsilon}{2}}\right\rceil\right)\right) = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_{\left\lceil n^{1-\frac{\epsilon}{2}}\right\rceil}]$, $R\left(\boldsymbol{n}, \log_2\left(\left\lceil n^{1-\frac{\epsilon}{2}}\right\rceil\right)\right) = [\boldsymbol{n}_1, \boldsymbol{n}_2, ..., \boldsymbol{n}_{\left\lceil n^{1-\frac{\epsilon}{2}}\right\rceil}]$ into equal length segment .

from (32), we can see that

$$\boldsymbol{y}_i = \mathbf{P}_{\left\lceil n^{\frac{\epsilon}{2}}\right\rceil}\boldsymbol{x}_i + \boldsymbol{n}_i = \mathbf{M}_{\left\lceil n^{\frac{\epsilon}{2}}\right\rceil}\boldsymbol{x}_i + \boldsymbol{n}_i, \forall i$$

2) Rounding Step: For each $\boldsymbol{y}_i$, we first do rounding, and then apply decode algorithm for noiseless code mentioned in section 4 of [11].

Since we do rounding first, if $\|\boldsymbol{n}_i\|_\infty < \frac{1}{2}$, then after rounding, the noisy part in $\boldsymbol{y}_i$ will vanish, and so the decoding result $\hat{\boldsymbol{x}}_i = \boldsymbol{x}_i$, hence, for those i such that $\hat{\boldsymbol{x}}_i \neq \boldsymbol{x}_i$, $\|\boldsymbol{n}_i\|^2 \geq \frac{1}{4}$. Combine with (34), the number of segments that possible wrong is smaller than

$$\frac{\left\|R\left(\boldsymbol{n}, \log_2\left(\left\lceil n^{1-\frac{\epsilon}{2}}\right\rceil\right)\right)\right\|^2}{\frac{1}{4}} = o\left(n^{2\delta+\frac{\epsilon}{2}}\right)$$

Since there are $\left\lceil n^{\frac{\epsilon}{2}}\right\rceil$ bits in each segment, the total number of error bits must smaller than

$$o\left(n^{2\delta+\frac{\epsilon}{2}}\right)\left\lceil n^{\frac{\epsilon}{2}}\right\rceil = o\left(n^{2\delta+\epsilon}\right) = o\left(n^{\kappa}\right)$$

Finally, in each deconstruction step we takes $O\left(\frac{n}{\log_2(n)}\right)$ operations, and we do $\log_2\left(\left\lceil n^{1-\frac{\epsilon}{2}}\right\rceil\right)$ times deconstruction, so the decode complexity in deconstruction step is $O(n)$. For the rounding step, it's easy to check that the decoding complexity for each data segment x_i is $O\left(\left\lceil n^{\frac{\epsilon}{2}}\right\rceil\right)$, and there are totally $\left\lceil n^{1-\frac{\epsilon}{2}}\right\rceil$ segments, hence the total decoding complexity for rounding step is $O(\bar{n}) = O(n)$, so the total decoding complexity for basic code is $O(n)$.

E. Proof of Theorem 4.4

By the property of $\mathbf{B}^{4n^\lambda \log(n)\times n}$ and the fact that $\sum_{i=1}^{4\log(n)} \mathbf{B}_i = \mathbf{B}^{4n^\lambda \log(n)\times n}$, we get

$$\begin{aligned}
&\forall \boldsymbol{x}, \boldsymbol{x}' \in \{0,1\}^{n\times 1}, \|\boldsymbol{x}\|_1, \|\boldsymbol{x}'\|_1 < l_n, \\
&\exists i \text{ such that } |\mathbf{B}_i\left(\{j : \boldsymbol{x}_j \neq \boldsymbol{x}'_j\}\right)| > \frac{\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_0}{6}}{4\log(n)} = \frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_0}{24\log(n)}
\end{aligned}$$

By pigeonhole principle, the number of index z such that $|\{j : \mathbf{B}_i(j) = z\}| \geq 48\log(n)$ is smaller than $\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_0}{48\log(n)}$, which implies the number of index z such that $|\{j : \mathbf{B}_i(j) = z\}| < 48\log(n)$ is greater than $\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_0}{24\log(n)} - \frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_0}{48\log(n)} = \frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_0}{48\log(n)}$. Note that in the construction of $\mathbf{D}_i$, we use BCH code check matrix with code distance $48\log(n)$ as our building block. Hence for these $\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_0}{48\log(n)}$ numbers of z, define $\boldsymbol{x}_z$ be a ternary vector such that $\boldsymbol{x}_z(j) = (\boldsymbol{x}-\boldsymbol{x}')(j)$ if $\mathbf{B}_i(j) = z$ and $\boldsymbol{x}_z(j) = 0$ otherwise. Then $\mathbf{D}_i\boldsymbol{x}_z \geq 1$ if $\boldsymbol{x}_z \neq \mathbf{0}$, combine with the property of Hadamard matrix $\left(\forall \boldsymbol{x} \in R^n, \|\mathbf{H}_n\boldsymbol{x}\|_2^2 = n\|\boldsymbol{x}\|_2^2\right)$, we get

$$\|\mathbf{D}_i(\boldsymbol{x}-\boldsymbol{x}')\|_2^2 \geq \frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_0}{48\log(n)} 4n^\lambda \log(n) = \frac{n^\lambda \|\boldsymbol{x}-\boldsymbol{x}'\|_0}{12}$$

By the construction of $\mathbf{D}_i$, the height of $\mathbf{D}_i$ is smaller than $4n^\lambda\log(n)48\log(n)^2 = 192n^\lambda\log(n)^3$, since height of $\mathbf{H}_{4n^\lambda\log(n)}$ is $4n^\lambda\log(n)$ and height of $C^n_{48\log(n)}$ is smaller than $48\log(n)^2$. Then

$$\begin{aligned}
\|\mathbf{Q}(\boldsymbol{x}-\boldsymbol{x}')\|_\infty \geq \|\mathbf{D}_i(\boldsymbol{x}-\boldsymbol{x}')\|_\infty &\geq \sqrt{\frac{\|\mathbf{D}_i(\boldsymbol{x}-\boldsymbol{x}')\|_2^2}{192n^\lambda\log(n)^3}} \\
&\geq \sqrt{\frac{\frac{n^\lambda\|\boldsymbol{x}-\boldsymbol{x}'\|_0}{12}}{192n^\lambda\log(n)^3}} = \frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_2}{48\log(n)^{\frac{3}{2}}}
\end{aligned}$$

The proof is complete. It remains to construct $B^{4n^\lambda\log(n)\times n}$. Although we can't construct $B^{4n^\lambda\log(n)\times n}$ explicitly, we can show the existence of such matrix by a probability method.

Theorem A.1: There exists a binary matrix $B^{4n^\lambda\log(n)\times n}$ with each column vector $4\log(n)$-sparse such that for any $1 \leq d \leq n^\lambda$ column vector $\boldsymbol{c}_{i_1}, ..., \boldsymbol{c}_{i_d}$, $\|V_{\{i_1,...,i_d\}}\| := \|\bigcup_{j=i_1,...,i_d} \boldsymbol{c}_j\|_0 \geq \frac{1}{6}d$.

Proof of Theorem A.1 is given in Appendix F of the extended version.

F. Proof of Theorem A.1

We use probabilistic method to proof this claim, let each element in $\mathbf{B}^{4n^\lambda \log(n)\times n}$ be i.i.d bernoulli $1/n^\lambda$ random variable, then the probability that the matrix $\mathbf{B}^{4n^\lambda \log(n)\times n}$ failed to have the desired property would be

$$
\begin{aligned}
P_f \leq^{(a)} & \sum_{S\subset[1:n],|S|\leq n^\lambda} Pr(\|V_S\| < \frac{1}{6}|S|) =^{(b)} \sum_{1\leq d\leq n^\lambda} \sum_{S\subset[1:n],|S|=d} \\
& Pr\left(BIN\left(4n^\lambda\log(n),(1-\frac{1}{n^\lambda})^d\right) > 4n^\lambda\log(n) - \frac{1}{6}d\right) \\
\leq^{(c)} & \sum_{1\leq d\leq n^\lambda} \sum_{S\subset[1:n],|S|=d} Pr\left(BIN\left(4n^\lambda\log(n),\frac{\frac{1}{3}d}{n^\lambda}\right) < \frac{1}{6}d\right) \\
= & \sum_{1\leq d\leq n^\lambda} \sum_{S\subset[1:n],|S|=d} Pr\left(e^{tBIN\left(4n^\lambda\log(n),\frac{\frac{1}{3}d}{n^\lambda}\right)} > e^{\frac{td}{6}}\right), t<0 \\
\leq^{(d)} & \sum_{1\leq d\leq n^\lambda} \sum_{S\subset[1:n],|S|=d} \frac{\left(\frac{d}{3n^\lambda}e^t + \left(1-\frac{d}{3n^\lambda}\right)\right)^{4n^\lambda\log n}}{e^{\frac{td}{6}}} \leq^{(e)} \sum_{1\leq d\leq n^\lambda} \sum_{S\subset[1:n],|S|=d} \frac{e^{\frac{4dn^\lambda\log(n)\left(e^t-1\right)}{3n^\lambda}}}{e^{\frac{td}{6}}} \\
=^{(f)} & \sum_{1\leq d\leq n^\lambda} \sum_{S\subset[1:n],|S|=d} e^{-\left(\frac{4d\log n}{3}-\frac{d}{6}-\frac{d\log(8\log(n))}{6}\right)} \leq^{(g)} \sum_{1\leq d\leq n^\lambda} n^d e^{-\frac{49}{48}d\log(n)}, n\geq 8 = o(1)
\end{aligned}
$$

(a) follows from union bound. (b) follows from the fact that V_S is the boolean sum of $\boldsymbol{c}_i, i\in S$.(c) follows from change the parameter of binomial distribution and the fact that $\forall 1\leq d\leq n^\lambda, (1-\frac{1}{n^\lambda})^d \leq 1-\frac{d}{3n^\lambda}$.$(d)$ follows from markov inequality. (e) follows from $1+x\leq e^x$. (f) follows from choose $t=-\log(8\log(n))$. (g) follows from upper bound the number of possible S with $|S|=d$ and the fact that $\frac{4d\log n}{3}-\frac{d}{6}-\frac{d\log(8\log(n))}{6} > \frac{49}{48}d\log(n), n\geq 8$.

Since $P_f = o(1)$, the matrix $B^{4n^\lambda\log(n)\times n}$ we desired must exists.