A framework for generalized group testing with inhibitors and its potential application in neuroscience

Thach V. Bui*, Minoru Kuribayashi†, Mahdi Cheraghchi§, and Isao Echizen††

*SOKENDAI (The Graduate University for Advanced Studies), Hayama, Kanagawa, Japan
†Graduate School of Natural Science and Technology, Okayama University, Okayama, Japan
‡Department of Computing, Imperial College London, UK
§Graduate School of Informatics, Tokyo, Japan

bvthach@nii.ac.jp kminoru@okayama-u.ac.jp

Abstract

The main goal of group testing with inhibitors (GTI) is to identify a small number of defective items and inhibitor items in a large set of items. A test on a subset of items is positive if it satisfies some specific property. Inhibitor items cancel the effects of positive items, which often make the outcome of a test containing defective items negative. Different GTI models can be formulated by considering how specific properties have different cancellation effects. This work introduces generalized GTI (GGTI) in which a new type of items is added, i.e., hybrid items. A hybrid item plays the roles of both defective items and inhibitor items. Since the number of GGTI models is large (at least 96), we introduce a framework for classifying all types of items non-adaptively, i.e., all tests are designed in advanced. We then explain how GGTI can be used to classify neurons in neuroscience. Finally, we optimize the construction of disjunct matrices, which are an important tool in GTI.

I. INTRODUCTION

A. Group Testing with inhibitors

GROUP TESTING: Identifying a small group of items satisfying a specific property \( \Upsilon \) in a colossal group of \( n \) items is the main problem in group testing. Such items are usually referred to as defective items, and the other items are usually referred to as negative items. The classification of group testing depends on the classification of defective items. Suppose that there are \( n \) items indexed from 1 to \( n \) and that the defective set \( D \subset [n] = \{1, \ldots, n\} \). A test on a subset of \([n]\) is designed to determine whether the subset satisfies \( \Upsilon \). If \( \Upsilon \) is satisfied, the test outcome is positive. Otherwise, the test outcome is negative.

In general, how \( D \) is determined to be a defective set and how members of the defective set \( D \) present in a test defines the test outcome. Here we assume that the upper bound of the cardinality of \( D \) is known, i.e., \( |D| \leq d \ll n \). The classifications of the defective items are as follows:

- Complex group testing \( \rightarrow \) Threshold group testing \( \rightarrow \) Classical group testing
- (Complex defectives) \( \rightarrow \) (Threshold defectives) \( \rightarrow \) (Classical defectives)

There are two main approaches to test design: adaptive and non-adaptive. In adaptive design, there are several stages of testing and the outcome of a test depends on the outcomes of previous tests. This approach usually achieves the minimum number of tests needed; the defective items are finally revealed by the last test. However, since there are several stages, it generally consumes lots of time. The non-adaptive design approach reduces the testing time because all tests are independent and designed a priori. However, this approach generally requires a larger number of tests.

Another important concern that should be considered is noise in the test outcome. The number of tests required in a noisy setting is usually larger than that in a noiseless setting.

The procedure for obtaining outcomes by testing subsets of items is called encoding and that for classifying the items from the test outcomes is called decoding. It is desirable to minimize the number of tests for the encoding procedure and minimize the time required for classifying the items. Other criteria are also considered for specific problems. The details of various types of group testing will be addressed in the following sections.

1) Classical group testing: In classical group testing (CGT), \( D \) is a defective set if the outcome of a test containing at least one item in \( D \) is positive, and negative otherwise. CGT has been intensively studied since its inception [1]. With adaptive design, the number of tests is \( O(d \log n) \) [2]. However, this design is problematic because it can take too much time to carry out the tests. With non-adaptive design, the number of tests is \( O(d^2 \log n) \) [2], [3]. Porat and Rothschild [4] first proposed explicit nonadaptive schemes to achieve this bound. However, there is no sublinear decoding algorithm associated with their schemes. For exact reconstruction in a noisy setting, Ngo et al. [5] proposed a scheme for identifying up to \( d \) defective items in
When the number of defective items is known, e.g., when \( TGT(O) \) w.h.p. Using numerical results, Bui et al. [8] recently showed that items in items. The most useful tool used to perform GTI is the (For decoding, there are also two approaches: 1) identify defective items only and 2) identify both defective items and inhibitor items. Considering exactly how many defective items there are in a test or their types. In this model, we have three sets out of three types of inhibitors: classical (dictator) inhibitors, threshold inhibitors, and complex inhibitors. We specify them without items. The inhibitor set is denoted as \( H \) defines the type of inhibitor item. An item is considered to be an inhibitor if it interferes with the identification of the defective items. The two parameters \( D \) and \( u \) consider the number of defectives items. Chin et al. [23] improve this bound to \( D \) disjunct matrix (defined later) can be used to identify complex defective items.

In threshold group testing (TGT), given two integer parameters \( d \) and \( u \) and \( Fu \) use condition on the number of defectives items. Damaschke [13] introduced TGT in 2006. By using \( \binom{n}{u} \) non-adaptive tests, he showed that a set of positive items could have up to \( g \) false positives and \( g \) false negatives. The number of tests was then decreased to \( O(d^{g/2} \log d \cdot \log \frac{g}{e}) \) by Cheraghchi [14]. When the number of defective items is known, e.g., \( d \), the number of tests can be reduced to \( O(d^{g/2} \log d) \) for \( g = 0 \) [15] or \( O(g^2 \log d) + O(d \log \frac{1}{e}) \) for \( e > 0 \) [16]. Although the number of tests is low, this approach is rarely applied because of the condition on the number of defectives items.

Most work on TGT has focused on the number of tests; there has been relatively little on the decoding procedure. Chen and Fu [17] use \( O \left( \frac{z^{d+u-f+1}}{\log d+u} \right) \) tests to find defective items in time \( O(n^c \log n) \) when there were at most \( z \) erroneous outcomes. Since the number of tests and the decoding time become huge as \( n \) increases, this approach is mostly impractical. Chan et al. [18] presented a randomized algorithm for finding defective items with \( O \left( \log^c d \cdot \sqrt{\log n} \right) \) tests in time \( O(g^2 n \log n + n \log \frac{1}{e}) \) given that the number of defective items is exactly \( d \) and \( u = o(d) \). Again, these conditions are too strict for practical applications, and the cost of decoding increases with \( n \). Bui et al. [19] recently proposed a scheme for finding up to \( d \) defective items using \( t = O \left( \left( \frac{z}{u} \right)^u \left( \frac{d}{u} \right)^{d-u} d^3 \log n \cdot \log \frac{g}{e} \right) \) tests in sub-linear time \( O(t \times poly(d, \log n)) \) when \( g = 0 \).

3) Complex group testing: In complex group testing (CmplxGT), \( D \) is a defective set if there are \( c \) smaller subsets \( D_1, \ldots, D_c \), such that:

- \( D = D_1 \cup \ldots \cup D_c \).
- Any \( D_a \) is the defective set in \( TGT(\ell_a, u_a) \), where \( 0 \leq \ell_a < u_a \leq u \leq d \) for \( a = 1, \ldots, c \).

When every defective subset of \( D \) has no gap, CmplxGT is called complex group testing without gap. When \( c = 1 \), CmplxGT reduces to TGT. When \( c = 1 \) and \( u_1 = 1 \), CmplxGT reduces to CGT. To avoid confusion with the definitions of defective items given in sections 1-A1 and 1-A2, we call defective items in CmplxGT complex defective items.

CmplxGT originated in molecular biology [20]. In this setting, \( \ell_a = u_a - 1 = |D_a| - 1 \). Chen et al. [21] restated this problem as complex group testing. There has been some work on CmplxGT [21–23]. It was found that an \( (d, r; z) \)-disjunct matrix (defined later) can be used to identify \( D \). In one such work [21], the number of non-adaptive tests is \( O \left( \left( \frac{z}{u} \right)^u \left( \frac{d}{u} \right)^{d-u} \log \frac{g}{e} \right) \), where \( z \) is the maximum number of erroneous outcomes. Without considering errors in the test outcomes, Chin et al. [23] improve this bound to \( O \left( (c + d) \frac{c^2 + d + 1}{c} \log n / (c^2 d^2) \right) \). These bounds increase as \( u \) or \( d \) increases.

INHIBITORS: Recent advances in the definition of group testing have added a new type of item: inhibitors. The manner in which the outcome of a test is positive defines the type of defective item, and the manner in which the outcome is negative defines the type of inhibitor item. An item is considered to be an inhibitor if it interferes with the identification of the defective items. The inhibitor set is denoted as \( H \), where \( |H| \leq h \ll n \). Similar to the classification of defective items, there are three types of inhibitors: classical (dictator) inhibitors, threshold inhibitors, and complex inhibitors. We specify them without considering exactly how many defective items there are in a test or their types. In this model, we have three sets out of \( n \) items: defective set \( D (|D| \leq d \ll n) \), inhibitor set \( H \), and negative set \( [n] \setminus D \cup H \). There are three classifications of inhibitor items: Complex inhibitors reduce to Threshold inhibitors reduce to Classical inhibitors.

For testing design, researchers often use a non-adaptive design to reduce testing time in group testing with inhibitors (GTI). For decoding, there are also two approaches: 1) identify defective items only and 2) identify both defective items and inhibitor items. The most useful tool used to perform GTI is the \( (d, u; z) \)-disjunct matrix (defined later). Here we present a more efficient construction of this matrix.
4) Classical (dictator) inhibitors: In group testing with classical inhibitors (GTDI), \( H \) is an inhibitor set if the outcome of a test containing at least one item of \( H \) is negative regardless of how many defective items are in the test \([22, 24\text{--}27]\). Let \( z \) be the maximum number of erroneous outcomes and \( \lambda = (d + h) \log n / W((d + h) \log n) + z \), where \( W(x) = \Theta(\log x - \log \log x) \). To identify defective items only, Chang et al. \([22]\) proposed a scheme using \( O((d + h + z)^2 \log n) \) tests in time \( O((d + h + z)^2 n \log n) \). For identifying both defective items and inhibitors, they proposed a scheme using \( O((d + h)^3 n \log n) \) tests in time \( O((d + h)^3 n \log^2 n) \). Without considering erroneous outcomes, Ganesh et al. \([26]\) use \( O((d + h) \log n) \) tests to identify defective items in time \( O((d + h)n \log n) \) by using a probabilistic scheme. It took \( O((d + h^2)n \log n) \) tests to identify both defective and inhibitor items in time \( O((d + h^2) n \log n) \). Bui et al. \([28]\) recently proposed a scheme that uses \( \Theta(\lambda^2 \log n) \) tests to identify only the defective set \( D \) in time \( O(\lambda^4 \log n / (d + h)^2) \). They also proposed another scheme for identifying both defective and inhibitor items in time \( O(d\lambda^6 \times \max \{\lambda^1 / (d + h)^2, 1\}) \) using \( O(\lambda^3 \log n) \) tests.

5) Threshold inhibitors: In group testing with threshold inhibitors (GGTI), \( H \) is an inhibitor set if the following statements hold:

- If the number of inhibitors in a test is less than \( u_i \), the test outcome is negative.
- If the number of inhibitors in a test is equal to \( l_i \), the test outcome depends only on the type and number of defective items.
- If the number of inhibitors in a test is more than \( l_i \) and less than \( u_i \), the test outcome depends on the type of defective items, the number of defective items, and the number of inhibitors in the test.

This threshold inhibitor model is denoted as GGTI \((li, ui)\). The two parameters, \( li \) and \( ui \), are the lower threshold and upper threshold. A gap is denoted as \( gi \), \( li \) and \( ui \), the test outcome depends only on the type and number of defective items.

Two previous works \([22, 29]\) specified GGTI with a formal definition of defective items. Both assumed that there is no gap in GGTI \((li, ui)\), i.e., \( gi = 0 \), and that there are up to \( e \) erroneous outcomes. The first considered two models of defective items: CGT and ComplexGT without a gap. When \( D \) is the classical defective set, there is a non-adaptive algorithm that can be used to classify all items by using an \((d + h - ul + 1, ul + 1; 2e + 1)\)-disjunct matrix. When \( D \) is the complex defective set as in section \([2A]\), things become complicated. Complex defective items could be identified only by using an \((d + h - ul + 1, ul + 1; 2e + 1)\)-disjunct matrix. The second work considered \( D \) to be the defective set in TGT discussed in section \([A2]\) and used an \((d + h - \ell - ul + 1, ul + 1; 2e + 1)\)-disjunct matrix to identify all defective items.

6) Complex inhibitors: In group testing with complex inhibitors (GTCI), \( H \) is an inhibitor set if there exists \( ci \) smaller subsets \( H_1, \ldots, H_c \), such that:

- \( H = H_1 \cup \ldots \cup H_c \).
- Any \( H_a \) is the inhibitor set in GGTI \((li_a, ui_a)\), where \( 0 \leq li_a < ui_a \leq h \) for \( a = 1, \ldots, ci \).

When \( ci = 1 \), GTCI reduces to GGTI. When \( ci = 1 \) and \( ui_1 = 1 \), GTCI reduces to GTDI. To avoid confusion with the inhibitor items defined in sections \([A4]\) and \([A5]\) we call defective items in GTCI complex inhibitor items.

There has been only one work dealing with complex inhibitors up to date \([30]\). The authors considered two error-tolerant models: CGT with complex inhibitors and TGT with complex inhibitors. Specifically, each small subset \( H_a \) is the inhibitor set in GGTI \((ui_a - 1, ui_a)\) for \( a = 1, \ldots, ci \). Their ultimate goal was to identify the defective \( D \) efficiently while minimizing the number of tests. They could not identify both defective items and inhibitors items. Let \( mi = \sum_{a=1}^{ci} ui_a \), \( ki = mi - ci + 1 \), and let \( z \) be the maximum erroneous outcomes. In CGT (TGT, resp.) with complex inhibitors, \( D \) is defined as in section \([A1]\) \([A2]\) (resp.). For both models, \( D \) can be recovered using \( O(z \log n \cdot \exp(d, mi, ki, h)) \) tests in time \( O(n^{mi+u} \log n) \).

7) Hybrid items: To generalize GTI, we introduce a new type of item: hybrid items. A hybrid item can be either defective or inhibitor. Under certain conditions, it is defective (inhibitory, resp.) because it satisfies the properties of a defective item (an inhibitory item, resp.). The formal definition of “certain conditions” is left for future work.

B. Action potentials in neuroscience

When you listen to a funny story, you often laugh. Your nervous system consisting of the central nervous system (CNS) and the peripheral nervous system (PNS) is responsible for this action. The CNS is located in the brain and spinal cord and is enclosed in bone. Neurons in the PNS travel through or lie on top of muscle, organ, and skin tissue. The primary purpose of the CNS is to organize and analyze signals from the sensory and motor neurons of the PNS, allowing us to observe and react to the environment. The central purpose of the PNS is to follow the commands of the CNS by changing motor output.

The nervous system is mainly regulated by three types of neurons: sensory neurons, interneurons, and motor neurons. Sensory neurons conduct signals from inside and outside the body such as those responsible for taste and vision to the CNS via stimulus receptors. Motor neurons convey signals from the CNS to the effector cells such as the muscles and glands. Finally, interneurons, which are distributed entirely within the CNS, interconnect sensory neurons and motor neurons. It is approximatley 86 billion neurons in a brain \([31]\). Now, going back to the mechanism underlying your laugh. The voice from the speaker reaches the sensory organs in your ear. The signals induced in those organs propagate to the sensory neurons, which connect with those in your spinal cord. The
sensory neurons then generate signals that propagate to the interneurons in your cerebral cortex. The brain cortex processes the signals received and then decides to move facial muscles via motor neurons. As a result, you laugh.

The signals in the laugh reaction chain are action potentials (APs) in the cells. There is an electrical potential difference between the inside of a cell and the surrounding environment. When a neuron is at rest, its membrane potential is about $-70\text{mV}$. When a neuron is active, an AP is caused by rapid depolarization of the membrane beyond threshold. The threshold is typically about $-55\text{mV}$. Therefore, an AP is an “all or none” phenomenon. This means that once the membrane has become depolarized and reaches the threshold, an AP will occur.

Next we describe how an AP is generated in terms of neuron interaction. A neuron contacts and communicates with other neurons by creating special sites called synapses using its axon and dendrites. Information, usually in the form of chemical substances called neurotransmitters, generally flows in one direction, from a source neuron to a target neuron. The source neuron is said to be presynaptic, and the target neuron is said to be postsynaptic. To generate an AP in a postsynaptic neuron, several presynaptic neurons of a target neuron release neurotransmitters. The type(s) of neurotransmitters released by a presynaptic neuron defines its neuron type in terms of neurotransmitters. Most excitations are generated by neurons releasing glutamate. Most inhibitions are generated by neurons releasing GABA. A more detailed explanation is available elsewhere [32].

Normally, there are two classifications of neurons based on the postsynaptic potential: excitatory postsynaptic potential (EPSP) and inhibitory postsynaptic potential (IPSP). A neuron is excitatory (inhibitory, resp.) if it makes EPSPs (IPSPs, resp.) at its postsynaptic neurons. Most neurons are unable to play both excitatory and inhibitory roles [33]. However, some neurons can be both excitatory and inhibitory [34], [35]. Such neuron are said to be hybrid. Moreover, some neurons may play neither role for a certain stimulus, so we have another type of neuron: negative neurons. We thus consider four types of neurons for a stimulus: excitatory neurons, inhibitory neurons, hybrid neurons, and negative neurons. Note that synaptic excitation and inhibition are inseparable events, even for the simplest sensory stimulus like a brief tone [36]. Any imbalance in synaptic excitation and inhibition may cause disorders such as schizophrenia and epilepsy. Another important point is that a postsynaptic neuron can receive neurotransmitters from more than one neuron, i.e., multiple synaptic potentials merge within one postsynaptic neuron. This process is called synaptic integration.

The main challenges related to APs are clarifying the mechanisms underlying how an AP is generated in a neuron and clarifying the mechanism of the interaction between neurons. A preliminary step in meeting these challenges is determining the neuron type corresponding to a stimulus. We have developed a scheme for classifying neurons for the stimulus.

C. Contributions

In this work, we make four contributions. First, we generalize group testing with inhibitors by introducing a new type of item: hybrid items that can play the role of a defective item or that of an inhibitor item. Second, we present an encoding/decoding framework for generalized group testing with inhibitors (GGTI). Third, we introduce a mapping between GGTI and neuron classification. Finally, to optimize the encoding of an instantiation of GGTI, we tackle the problem of constructing an $(d, u; z)$-disjunct matrix. In particular, the number of rows in this matrix is $O\left(\left(\frac{d'}{2e^{z+1/2}}\right)^{\kappa} \cdot z \cdot \frac{1}{p}\right)$, where $\kappa = \lceil \log_2 \frac{d+u}{u} \rceil$, $d' = \max\{d+u, u2^{\kappa}\}$, $1 < \frac{1}{p} < \frac{1}{4}$, and $z = \frac{1}{p} \prod_{i=1}^{\kappa} \ln \left(\frac{n}{p} \cdot 2^i\right)$.

This bound is significantly better than the well-known bound $O\left(\left(\frac{d+u}{u}\right)^{\kappa} \left(\frac{d+u}{d}\right)^{d} \frac{u}{d+u} \log \frac{n}{p+u}\right)$.

D. Applications

1) Neuroscience: Medicine: Stimuli can be generated both inside and outside the body. Our objective is to locate which neurons are responsible for sensing/responding to a particular stimulus and then determine the types of those neurons. There are numerous potential applications of this capability. We mention only a few of them here. When a patient undergoes major surgery, the patient’s body is usually completely anesthetized so that the patient is unconscious before and during surgery. The main post-operative task is to restore the patient’s consciousness. If we could localize the neurons responsible for reaction in the part of the patient’s body that is the surgical target, we could anesthetize only those neurons. This might obviate the need to restore consciousness because it might be possible for the patient to remain conscious during surgery. This would lessen the risk posed by surgery. Another potential application is the treatment of such disorders as acrophobia, schizophrenia, and epilepsy. Identification of the neurons responsible for these disorders would enable application of personalized treatment. A final example is the creation of a brain stimulus paradigm and then identifying the mechanism underlying that stimulus.

Spike neural networks: Spike neural networks [37], [38] have been intensively studied to mimic how a brain works. They have been applied in a wide range of machine learning, including computer vision [39] and pattern recognition [40]. Lynch et al. [41] presented a computational model of a neuronal network for investigating how inhibitory neurons work in the brain. They considered a network consisting of $n$ input neurons and $n$ corresponding output neurons. They estimated how many inhibitory neurons needed to be added to the network so that a single neuron fires an AP while the other neurons did not fire APs. Identifying the firing neuron was their ultimate goal. It turned out to be the problem of identifying a firing neuron by minimizing the number of inhibitory neurons in a set of inhibitory and excitatory neurons via testing neurons one by one. In
their work, given \( n \) neurons in total, \( O(\theta) \) adaptive rounds, i.e., \( O(\theta n) \) tests, are needed if there are \( O(\theta \log^{1/\theta} n) \) inhibitory neurons. In their model, the authors placed many constraints on the network. First, only one neuron could fire an AP. Second, the number of inhibitory neuron was constrained by \( n \) and a constant. Third, each neuron function was a probabilistic threshold unit, spiking with a given probability. Therefore, the two types of neurons become at most three types of neurons in a round: inhibitory neurons, excitatory neurons, and negative neurons. Fourth, each connection in the network had a weight. Their objective was to find the excitatory neuron firing an AP corresponding to the chains of interaction between neurons in the network.

The model proposed by Lynch et al. [41] was set up so that each type of neurons had a unique set of specific properties. They used it to determine whether a neuron in the network fires an AP. This is an unusual way to illustrate how an AP is generated in the brain. Even if there is a neuron that fires an AP, it is uncertain that their model accurately represented the firing of a real one in the brain. Our approach is more natural. We first identify the types of neurons in the brain (which is done here). Then we can build the connections between these neurons (left for future work). In our model, there are four types of neurons in the network: excitatory neurons, inhibitory neurons, hybrid neurons, and negative neurons. Moreover, our proposed scheme is non-adaptive.

2) Learning a hidden hypergraph: Angluin and Chen [42] described the problem of learning a hidden hypergraph as follows. Consider a set of \( n \) items and a given family \( C \) of subsets of \([n]\). The objective is to identify an unknown family \( D = \{D_a\} \) from the given family \( C \), where \( a = 1, \ldots, c \), \( |D_a| \leq u \), \( |D| \leq d \), and a certain property holds if all the members in each \( D_a \) appear in a test. Precisely, a test is positive if it contains all members of any \( D_a \), and negative otherwise.

The complex \( C \) is viewed as a hypergraph with the vertex set of \( n \) items. Every \( D_a \) is considered to be an edge of the hypergraph, and \( D \) is a hidden graph with a size up to \( d \). The only operation to be carried out is to test whether a set of \( n \) vertices includes an edge of \( D \). The goal is to identify the hidden subgraph \( D \) in the given hypergraph \( C \) with the minimum number of tests. This problem turns out to be an instance of complex group testing and can be resolved by using an \((d, u; z)\)-disjunct matrix. The number of tests, i.e., the number of rows in the matrix, is \([z - 1]/2\) is the maximum number of erroneous outcomes.

D'yachkov et al. [43] proposed two adaptive schemes for tackling this problem. They assumed that the number of items goes to infinity. For multistage testing, they proposed an algorithm that takes at most \( O(du \log n) \) tests and stages to identify the hidden hypergraph. If the number of stages is two, the number of tests remains the same, but their design is random. This setting is restrictive in practice because the number of items is usually sufficient large, but not extremely large such as infinity. Moreover, the error-tolerance was not considered. Chen et al. [21] demonstrated that a one-stage is feasible by using an \((d, u; z)\)-disjunct matrix. The number of tests, i.e., the number of rows in the matrix, in this design is \( O \left( \left( \frac{d}{u^{\frac{1}{d}} + 1} \right)^\kappa \cdot u \cdot \frac{1}{p} \right) \), where \( \kappa = \lceil \log_2 \frac{d + u}{u} \rceil \), \( d' = \max \{d + u, u 2^n\} \), \( 1 < \frac{1}{p}, \frac{1}{d'} \), and \( z = \frac{p}{d'} \cdot \prod_{i=1}^\kappa \ln \left( \frac{p}{d'} \cdot 2^i \right) \).

E. Techniques

Our key techniques involve two concomitant variates: the tensor product and the divide-and-conquer strategy. The techniques are applied to the encoding and decoding of GGTI and the construction of \((d, u; z)\)-disjunct matrices. The generalization of group testing with inhibitors and its mapping to neuron classification are done in terms of abstract formalization. Since the divide-and-conquer strategy depends on the problems, we only define the tensor product here. Let \( \otimes \) be the tensor product notation. Given an \( f \times n \) matrix \( A = (a_{ij}) \) and an \( s \times n \) matrix \( S = (s_{ij}) \), the tensor product of \( A \) and \( S \) is defined as

\[
\mathcal{R} = A \otimes S := \begin{bmatrix}
S \times \text{diag}(A_{1,*}) \\
\vdots \\
S \times \text{diag}(A_{f,*})
\end{bmatrix} = \begin{bmatrix}
[a_{11}S_1 & \ldots & a_{1n}S_n] \\
\vdots \\
[a_{f1}S_1 & \ldots & a_{fn}S_n]
\end{bmatrix},
\]

where \( \text{diag}(\cdot) \) is the diagonal matrix constructed from the input vector, and \( \mathcal{A}_{h,*} = (a_{h1}, \ldots, a_{hn}) \) is the \( h \)th row of \( A \) for \( h = 1, \ldots, f \). The size of \( \mathcal{R} \) is \( r \times n \), where \( r = f \times s \).

II. Preliminaries

For consistency, we use capital calligraphic letters for binary matrices, non-capital letters for scalars, capital letters for sets, and bold letters for vectors. Here are some notations used:

1) \( n, \mathbf{x} = (x_1, \ldots, x_n)^T \): number of items and representation vector of \( n \) items.
2) \( T_j, T_i, T_{i,*}, T_{j,*}, \mathcal{A}_{i,*}, \mathcal{M}_{i,*}, \mathcal{M}_j \): column \( j \) of matrix \( T \), row \( i \) of matrix \( T \), row \( i \) of matrix \( \mathcal{G} \), row \( i \) of matrix \( \mathcal{M} \), and column \( j \) of matrix \( \mathcal{M} \), respectively.
3) \( \text{diag}(\mathcal{G}_{i,*}) = (g_{1i}, \ldots, g_{ni}) \): diagonal matrix constructed by input vector \( \mathcal{G}_{i,*} \).
A. Measurement matrix

As we outlined in section [1], there are two main approaches in testing design: adaptive and non-adaptive. In adaptive design, there are several stages of testing, and the outcome of a test depends on the outcomes of previous tests. Since adaptive design is time consuming, it is not preferred in practice. Therefore, we focus on non-adaptive design in which all tests are independent and designed a priori. The tests can be represented as follows. Let $n$, $D$, $H$, and $B$ be the number of items, the defective set, the inhibitor set, and the hybrid set, where $|D| \leq d$, $|H| \leq h$, $|B| \leq b$, and $1 \leq d + h + b \leq n$. Let $x = (x_1, \ldots, x_n)^T$ be the representative vector of $n$ items, where $x_j = 0$ means that item $j$ is negative and that item $j$ is either defective, inhibitory, or hybrid, otherwise. Note that we do not specify which values represent defective, inhibitor, and hybrid. For a $t \times n$ binary measurement matrix $T = (t_{ij})$, item $j$ is represented by column $T_j$, test $i$ is represented by row $i$, $t_{ij} = 1$ if and only if item $j$ belongs to test $i$, and $t_{ij} = 0$ otherwise.

Define $\text{supp}(v) = \{j \mid v_j \neq 0\}$ for any vector $v = (v_1, \ldots, v_n)$. Let $\text{test}(S)$ be the test on subset $S \subseteq [n]$. The outcome of the test is either positive (1) or negative (0) and depends on the definition of $D$, $H$, $B$, and $S$. The tests on $n$ items using $T$ is defined as

$$y = T \cdot x = \begin{bmatrix} \text{test(\text{supp}(T_{i,*}) \cap \text{supp}(x))} \\ \vdots \\ \text{test(\text{supp}(T_{r,*}) \cap \text{supp}(x))} \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_t \end{bmatrix},$$

where $y_i = \text{test(\text{supp}(T_{i,*}) \cap \text{supp}(x))}$. The procedure to get $y$ is called the encoding procedure. It includes the construction procedure, which is used to get measurement matrix $T$. The procedure to recover $x$ from $y$ and $T$ is called the decoding procedure. Our objective is to design measurement matrix $T$ with the small number of tests such that $x$ can be recovered efficiently when outcome vector $y$ is observed.

B. Disjunct matrices

The most useful measurement matrices in group testing with inhibitors are disjunct matrices. They were first introduced by Kautz and Singleton [44] as superimposed codes and then generalized by D’yachkov et al. [45] and Stinson and Wei [46]. They have also been given other names such as separating codes [47], [48]. Here we use “disjunct matrices”. The formal definition of a disjunct matrix is as follows.

**Definition 1.** An $m \times n$ binary matrix $M$ is called an $(d, r; z)$-disjunct matrix if for any two disjoint subsets $S_1, S_2 \subseteq [n]$ such that $|S_1| = d$ and $|S_2| = r$, there exists at least $z$ rows in which there are all 1’s among the columns in $S_2$ while all the columns in $S_1$ have 0’s, i.e., $\left| \bigcap_{j \in S_1} \text{supp}(M_j) \right| \cup \bigcup_{j \in S_1} \text{supp}(M_j) \geq z$.

III. GENERALIZED GROUP TESTING WITH INHIBITORS

A. Model

In this section, we generalize GTI by introducing a new type of item: hybrid items. In GTI, defective items only tend to make the outcome of a test on them positive, and inhibitor items only tend to make the outcome of a test on them negative. Hybrid items, however, can make the outcome of a test on them either positive or negative. The GGTI paradigm is illustrated in Figure[1]. It has encompassed two procedures: encoding and decoding. Encoding includes designing a measurement matrix by choosing the model and criteria and then obtaining the test outcomes. Decoding is the classification of items on the basis of the test outcomes.

The model and criteria should be selected so as to obtain a suitable measurement matrix. The factors to consider when selecting the model are the number of item types, the noise setting in the test outcomes, and the testing design. As discussed in section[1] there are three types of defective items, three types of inhibitor items, and at least one type of hybrid items. For the noise setting, there are two types: noisy and noiseless. Similarly, there are two types of testing design: adaptive and non-adaptive.

For non-adaptive design, the measurement matrix should be optimized after selecting a suitable model. As shown in Figure[1] six criteria are considered here: construction type, decoding time, number of tests, space to generate an entry, time to generate an entry, and gap-zeros. Construction type is usually random, i.e., the matrix is generated randomly, or non-random, i.e., the matrix is generated nonrandomly. The “gap-zeros” criterion is extremely helpful in biological applications. It would reduce time to pick items when making tests in a row. In particular, a $t \times n$ matrix $T$ has gap-zeros if for any permutation of the rows in $T$, $\sum_{i=1}^t \text{wt}(T_{i,*}, T_{i+1,*})$ is minimum, where $\text{wt}(u, v)$ is the Hamming distance of vectors $u$ and $v$. For example, if “classical defective”, “noisy setting”, “non-adaptive design”, and “decoding time” are selected, the model is the CGT model defined in section[1A].

A measurement matrix is ideal if it can be used to efficiently classify the given items in a noisy setting with a non-adaptive design, can be generated nonrandomly when there is little time and space to generate its entries, and has gap-zeros.
Let \( y_j \) means item matrix encoding and decoding for the GGTI paradigm should be considered. Matrices and models are extremely large, it would take time to design every measurement matrix. Therefore, a framework of encoding is to create a measurement matrix and then do tests on it to get outcomes. The matrix is created by selecting a model and criteria. If the background of an outer block is aqua, one of the sub-blocks must be selected. Specifically, the defective item type, noise setting, testing design, and criteria must be selected to create a measurement matrix. The inhibitor and hybrid item types can be selected corresponding with the problem.

**B. Encoding and decoding framework**

For paradigm illustrated in Figure[1] the minimum number of models is \( 3 \times 4 \times 2 \times 2 = 96 \) and the minimum number of possible measurement matrices is: \( (3 \times 4 \times 2 \times 2) \times (2^6 - 1) = 6,048 \). Since the minimum numbers of possible measurement matrices and models are extremely large, it would take time to design every measurement matrix. Therefore, a framework of encoding and decoding for the GGTI paradigm should be considered.

1) **Encoding procedure:** To design a measurement matrix, we introduce notation for a perfect pair of two matrices. Let \( v \) be an \( n \times 1 \) vector. Matrices \( G \) and \( M \) are considered to be a perfect pair if:

- There is an index set of rows of \( G \), denoted \( D' \), such that \( \text{supp}(G_{i,*}) \cap T \) satisfies property \( T_i \) for \( i \in D' \), and \( D \subseteq \bigcup_{i \in D'} \text{supp}(G_{i,*}) \cap T \). Set \( T = \{T_1, \ldots, T_{|D'|}\} \).

- There is an index set of rows of \( G \), denoted \( H' \), such that \( \text{supp}(G_{i,*}) \cap T \) satisfies property \( H_i \) for \( i \in H' \), and \( H \subseteq \bigcup_{i \in H'} \text{supp}(G_{i,*}) \cap T \). Set \( H = \{H_1, \ldots, H_{|H'|}\} \).

- There is an index set of rows of \( G \), denoted \( B' \), such that \( \text{supp}(G_{i,*}) \cap T \) satisfies property \( B_i \) for \( i \in B' \), and \( B \subseteq \bigcup_{i \in B'} \text{supp}(G_{i,*}) \cap T \). Set \( B = \{B_1, \ldots, B_{|B'|}\} \).

- If \(|v| \leq m_0\) and supp\((v)\) satisfies some property in \( T, H, \Phi \), or \( \Psi \), vector \( v \) can always be recovered from \( M \circ v \), and denoted \( v = \text{dec}(M \circ v, M) \).

- If \(|v| > m_0 \) or supp\((v)\) does not satisfy any property in \( T, H, \Phi \), or \( \Psi \), then \( \text{dec}(M \circ v, M) \) either cannot recover \( v \) or recovers a vector \( v' \neq v \), where \(|v'| \leq m_0\).

Note that in the above definition, matrix \( M \) can “handle” all three properties, \( T, H, \Phi \), and \( \Psi \). We thus create a measurement matrix \( T = G \otimes M \), where \( \otimes \) is defined in [1]. Let \( x = (x_1, \ldots, x_n)^T \) be the representation vector of \( n \) items, where \( x_j = 0 \) means item \( j \) is negative and item \( j \) is either defective, inhibitory, or hybrid, otherwise. Note that supp\((x) = T \). Then the outcome vector, \( y = T \circ x \), obtained by using \( x \) and \( T \) is
Let a hybrid neuron generates an AP, propagates that AP to another neurons, and inhibits other neurons from generating APs.

As described in section I, there are four types of neurons for a stimulus: excitatory, inhibitory, hybrid, and negative. A neuron generating an AP must be excitatory, inhibitory, or hybrid. Negative neurons do not generate APs. Therefore, we formally define four types of neurons:

1) A negative neuron does not generate an AP.
2) An excitatory neuron generates an AP and propagates that AP to another neurons.
3) An inhibitory neuron generates an AP and inhibits other neurons from generating APs.
4) A hybrid neuron generates an AP, propagates that AP to another neurons, and inhibits other neurons from generating APs.

Although it is difficult to have a hybrid neuron for a stimulus, a hybrid neuron can be detected by observing different stimuli. For some stimuli, a hybrid neuron plays the role of an excitatory neuron. For other stimuli, a hybrid neuron plays the role of an inhibitor.
Fig. 2: Illustration of generating an \((6, 2; e)\)-disjunct matrix. Suppose that \(u = 2\) and \(d = 8\). Then \(\kappa = \log_2 \frac{d}{u} = 2\). Therefore, we have to generate an \(\left(\frac{d}{2} = 4, \frac{d}{2} = 4; e_1\right)\)-regular matrix \(M_1\) and an \(\left(\frac{d}{2} = 2, u = \frac{d}{2} = 2; e_2\right)\)-regular matrix \(M_2\). Since we initialize \(M_1\), the eight items are split into two disjoint subsets at level 1. The cardinality is equal to \(d/2 = 4\). In other words, there are two rows such that, in each row, there are exactly \(d/2\) potential defective items. We recursively multiply \(M_1\) at level 1 by \(M_2\) by using the tensor product to obtain the final measurement matrix at level 2, \(M = M_1 \otimes M_2\). At this level, there are at least \(2^\kappa\) rows such that each row contains exactly \(u = \frac{d}{2}\) potential defective items. Moreover, the potential defective items in these rows comprise the full set of eight potential defective items. The result is that \(M\) is an \((d - u = 6, u = 2; e = (e_1 + 1)(e_2 + 1) - 1)\)-disjunct matrix.

B. Mapping between generalized group testing with inhibitors and classification of neurons

It is natural to do mapping between GGTI and neuron types. Excitatory neurons, inhibitory neurons, hybrid neurons, and negative neurons are represented by defective items, inhibitor items, hybrid items, and negative items in GGTI. Each type of neuron has corresponding sub-types of items in GGTI. However, because of the “all or none” characteristic of APs, the outcome of a test is either positive, i.e., an AP is generated, or negative, i.e., no AP is generated. It is equivalent to the case that there is no gap in GGTI for TGT, CmplXG, GGTI, and GTCl. Because excitation and inhibition are inseparable events in the brain and synaptic integration always occurs, the model selected must have defective and inhibitor items. The model selected for a neural network is transformed into a model in GTGI, as illustrated in Figure 1. A corresponding measurement matrix is then generated. The measurement matrix is then used to perform tests and obtain test outcomes. Finally, the type of each neuron is identified after performing a decoding procedure.

The actual design of a test for the brain is an open problem. Moreover, the method presented in Theorem 1 is simply a possible method for creating a measurement matrix. Concrete methods for generating a measurement matrix need to be developed.

V. IMPROVED CONSTRUCTION OF \((d, u; z)\)-DISJUNCT MATRIX

In this section, we present an efficient construction of an \((d, u; z)\)-disjunct matrix with the aid of regular matrices. Let \(M|_S\) be the \(m \times |S|\) submatrix of \(M\) formed by restricting matrix \(M\) to the columns picked by \(S\), where \(m\) is the number of rows in \(M\). Then we have:

**Definition 3** (Definition 6 [14]). Let \(n, d, e, u\) be non-negative integers where \(0 < u \leq d \leq n\). A binary matrix \(M\) with \(n\) columns is called threshold \((d, u; e)\)-regular if for every subset of columns \(S \subseteq [n]\) (called the critical set) and every \(Z \subseteq [n]\) (called the zero set) such that \(u \leq |S| \leq d, |Z| \leq |S|, S \cap Z = \emptyset\), there are more than \(e\) rows of \(M\) at which \(M|_S\) has weight exactly \(u\), (at the same rows) \(M|_Z\) has weight zero.

To reduce the number of rows in \((d, u; z)\)-disjunct matrices, we recursively use the tensor product for several regular matrices. Our objective is to generate an \((d - u, u; e)\)-disjunct matrix, as illustrated in Figure 2 for \(d = 8\) and \(u = 2\). Let \(\kappa = \log_2 \frac{d}{u} = 2\) be the number of regular matrices used in this idea. For any set \(S \subseteq [n]\) and \(|S| = 8\), our goal is to partition \(S\) into \(\frac{d}{u} = 4\) subsets such that the cardinality of each subset is \(u = 2\) and there are at least \(e\) rows in \(M\) that each contain only that subset. We use a recursive strategy to reduce the presence of items in \(S\) in some rows until the number of items in \(S\) in those rows is exactly \(u\).

This idea is formalized as the following theorem.

**Theorem 2.** Let \(2 \leq u \leq d\) be integers, \(\kappa = \lceil \log_2 \frac{d}{u} \rceil\), and \(d' = \max \{d, u2^\kappa\}\). Suppose that \(M_1\) is an \(m_i \times n\)

\[
\left(\frac{d'}{2^{(i+1)/2}}, \frac{d'}{2^i}; e_i = \Omega \left(\frac{p_i}{(1-p_i)^2} \ln \frac{n}{d'}\right)\right)\text{-regular matrix, where } p_i \in [0, 1] \text{ and } i = 1, 2, \ldots, \kappa.
\]

Then the matrix \(M = M_1 \otimes M_2 \otimes \ldots \otimes M_\kappa\) is an \((d - u, u; e)\)-disjunct matrix, where \(e = \prod_{i=1}^\kappa (e_i + 1)\). Moreover, the number of rows in \(M\) is

\[
m = \prod_{i=1}^\kappa m_i = O \left( \left(\frac{d'}{2^{(i+1)/2}}\right)^\kappa \prod_{i=1}^\kappa \ln \left(\frac{n}{d'} \cdot 2^i\right) \prod_{i=1}^\kappa \frac{1}{(1-p_i)^2}\right).
\]

When \(\log_2 \frac{d}{u}\) is an integer, \(m = O \left( \left(\frac{d'}{2^i}\right)^{\frac{1}{2} \log_2 \frac{d}{u}} \prod_{i=1}^{\log_2 \frac{d}{u}} \left(\frac{n}{d'} \cdot 2^{i}\right) \prod_{i=1}^{\log_2 \frac{d}{u}} \frac{1}{(1-p_i)^2}\right)\).
We thus conclude that we get

**Appendix**

**A. Proof of Theorem 2**

We have \( \text{supp}(\text{diag}(G_i) \times x) = \text{supp}(G_i) \cap \text{supp}(x) = \text{supp}(G_i) \cap T \). Therefore, for any \( i \in D' \), \( \text{diag}(G_i) \times x \) satisfies property \( T_i \) because of the definition of \( G \). Moreover, we have \( \text{supp}(\text{diag}(G_i) \times x) = |\text{supp}(G_i) \cap T| \leq m_0 \). Therefore, we get \( \text{dec}(y_i, M) = \text{diag}(G_i) \times x \). From the definition of \( G \), \( D \subseteq \bigcup_{i \in D} \text{supp}(G_i) \cap T = \bigcup_{i \in D} \text{supp}(\text{dec}(y_i, M)) \). Thus, performing Steps 2 to 4 in Algorithm 1 ensures that the set \( \mathcal{S} \) contains \( D \). Similarly, it can be proved that \( \mathcal{S} \) contains \( H \) and \( B \). We thus conclude that \( T = D \cup H \cup B \subseteq S \).

Because it takes time \( O(A) \) to perform Step 3 \( \{\text{dec}(y_i, M)\} \leq m_0 \) (the definition of \( M \) in Definition 2), and the number of loops in Step 2 is \( g \). We thus have \( |S| \leq g m_0 \) and the running time is \( O(gA) \). Therefore, the measurement matrix \( T \) can be used to identify \( D \cup H \cup B \) in time \( O(gA) \) with at most \( g \times \max\{d, h, b\} \) misclassified items in the set recovered.

**B. Auxiliary lemma**

Cheraghchi [14] showed that there is an efficient construction of a regular matrix:

**Lemma 1** (Lemma 14 [14]). For every \( p \in [0, 1) \) and integer parameter \( 1 \leq d \leq p \) exists an \( h \times n \) \((d, d, \Omega(ph/d))-regular matrix with probability \( 1-o(1) \), where \( h = O(d(\ln n)^2/(1-p)^2) \).

The connection of regular matrices with the special case of disjoint matrices is stated in the following lemma:

**Lemma 2.** Let \( p \in [0, 1) \), \( 0 \leq e \), and \( 1 \leq d \leq n \) be integers. Then any \((d, d; e \cdot d)-disjunct matrix is an \((d, d; e + 1 \cdot d)-disjunct matrix. The number of rows in these matrices is \( O(d(\ln n/d)/(1-p)^2) \).

**Proof.** We consider the \( h \times n \) \((d, d; e))-regular matrix \( \mathcal{M} \) defined in Definition 3. For any subset \( S, Z \subseteq [n] \) such that \( S \cap Z = \emptyset \) and \( |S| = |Z| = d \), there are more than \( e \) rows in \( \mathcal{M} \) in which \( \mathcal{M}|_S \) has a weight of exactly \( d \) and (in the same rows) \( \mathcal{M}|_Z \) has a weight of zero. Therefore, \( \mathcal{M} \) is an \((d, d; e + 1 \cdot d)-disjunct matrix. Moreover, from Lemma 3 \( h = O(d(\ln n/d)/(1-p)^2) \).

**C. Proof of Theorem 2**

We consider two cases for \( \log_2 \frac{d}{2} \): integer and non-integer. When \( \log_2 \frac{d}{2} \) is an integer, \( \kappa = \log_2 \frac{d}{2} \), and \( d' = \max\{d, u2^\kappa\} = d = u2^\kappa \). When \( \log_2 \frac{d}{2} \) is not an integer, \( \kappa = \lfloor \log_2 \frac{d}{2} \rfloor \). Note that \( 2^{\kappa-1} < \frac{d}{2} < 2^\kappa \). Therefore, we have \( d < d' = u2^\kappa < 2d \) and \( d' = \max\{d, u2^\kappa\} = u2^\kappa \). Then we generate an \((d' - u; u; e\cdot d')\)-disjunct matrix. Because \( d' - u > d - u \), an \((d' - u; u; e\cdot d')\)-disjunct matrix is also an \((d - u; u; e\cdot d')\)-disjunct matrix. For any case of \( \log_2 \frac{d}{2} \), we have \( d' = u2^\kappa \).

Let us consider any set \( S \subseteq [n] \) such that \|S\| = d'. Our objective is to partition \( S \) into \( \frac{d}{2} \) subsets such that the cardinality of each subset is \( u \) and there are at least \( e \) rows in \( \mathcal{M} \) that contains only that subset. Then \( \mathcal{M} \) is an \((d' - u; u; e\cdot d')\)-disjunct matrix.

Let \( S_1 \) and \( S_2 \) be two subsets of \( S \) such that \( S_1 \cup S_2 = S \) and \|S_1\| = \|S_2\| = \frac{|S|}{2} = \frac{d}{2} \). Since \( \mathcal{M}|_1 \) is an \((d' - u; u; e\cdot d')\)-regular matrix, there is a set \( E_1 (E_2) \) consisting of at least \( e_1 = 1 + \Omega(p_1 m_1/(d'/2)) \) rows in \( \mathcal{M}|_1 \) which all entries in \( \mathcal{M}|_1 \) are \( 1 \)s (have a weight of exactly \( d'/2 \)) and (in the same rows) all entries \( \mathcal{M}|_1 \) are \( 0 \)s (have a weight of zero), where \( m_1 = O((d'/2)\log(n/(d'/2))/(1-p_1)^2) \). At this level, denoted level 1, since we use only matrix \( \mathcal{M}_1 \), the original set \( S \) is divided into \( 2^1 \) subsets such that the cardinality of each subset is \( \frac{d}{2} \) and there are at least \( e_1 + 1 \) rows in \( \mathcal{M}_1 \) that contains only that subset. The number of rows in \( \mathcal{M}_1 \) is \( m_1 \).

We now consider to the product \( \mathcal{M}_1 \odot \mathcal{M}_2 \). Since there are two sets \( E_1 \) and \( E_2 \), we consider set \( E_1 \) while treating the remaining set the same as \( E_1 \). We now consider only the row for which the index belongs to \( E_1 \). For any \( k \in E_1 \), all entries in row \( k \) in \( \mathcal{M}_1 \) are \( 1 \)s. Because we apply the tensor product to \( \mathcal{M}_1 \) by using \( \mathcal{M}_2 \), the \( k \)th row in \( \mathcal{M}_1 \) (denoted \( \mathcal{M}_1(k,:) \)) becomes \( \mathcal{M}_1(k,:) \odot \mathcal{M}_2 \). Let \( S_1 \) and \( S_2 \) be two subsets of \( S \) such that \( S_1 \cup S_2 = S \) and \|S_1\| = \|S_2\| = \frac{|S|}{2} = \frac{d}{2} \). We have that all members of \( S_1 \) belong to \( \mathcal{M}_1(k,:) \) while the other members in \( S \setminus S_1 \) belong to that row and therefore the members of \( S \setminus S_1 \) do not. At this level (level 2), since we use two matrices \( (\mathcal{M}_1 \text{ and } \mathcal{M}_2) \).
the original set $S$ is divided into $2^2$ subsets such that the cardinality of each subset is $d_2'$, and there are at least $(e_1 + 1)(e_2 + 1)$ rows in $M_1 \otimes M_2$ that each contains only that subset. Because of the tensor product, the number of rows in $M_1 \otimes M_2$ is $m_1m_2$.

Recursively, at level $\tau$, we use $\tau$ matrices $M_1, \ldots, M_{\tau}$, the original set $S$ is divided into $2^\tau$ subsets such that the cardinality of each subset is $d_\tau'$, and there are at least $\prod_{i=1}^{\tau}(e_i + 1)$ rows in $M_1 \otimes M_2 \otimes \cdots \otimes M_{\tau}$ that each contains only that subset. Moreover, the number of rows in $M_1 \otimes M_2 \otimes \cdots \otimes M_{\tau}$ is $\prod_{i=1}^{\tau} m_i$. When $\tau = \kappa$, the original set $S$ is divided into $2^\kappa$ subsets such that the cardinality of each subset is $d_\kappa'$, and there are at least $\prod_{i=1}^{\kappa}(e_i + 1)$ rows in $M_1 \otimes M_2 \otimes \cdots \otimes M_{\kappa}$ that each contains only that subset. Moreover, the number of rows in $M_1 \otimes M_2 \otimes \cdots \otimes M_{\kappa}$ is $\prod_{i=1}^{\kappa} m_i$.

From Lemma 2, since $M_i$ is an $m_i \times n \left(\frac{d_i'}{2^i}, \frac{d_i'}{2^i}; e_i\right)$-regular matrix, the number of rows in $M_i$ is:

$$m_i = O\left(\frac{d_i'}{2^i} \ln \left(\frac{n}{d_i'/2^i}\right) \cdot \frac{1}{(1-p_i)^2}\right) = O\left(\frac{d_i'}{2^i} \ln \left(\frac{n}{d_i'} \cdot 2^i\right) \cdot \frac{1}{(1-p_i)^2}\right).$$

Moreover, we also imply that $e_i = \Omega \left(\frac{p_i}{(1-p_i)^2} \cdot \ln \left(\frac{n}{d_i'/2^i}\right)\right)$. Then we have

$$m = \prod_{i=1}^{\kappa} m_i = O\left(\frac{(d')^\kappa}{2^{\kappa \log_2 \frac{d'}{u}}} \prod_{i=1}^{\kappa} \ln \left(\frac{n}{d_i'} \cdot 2^i\right) \cdot \prod_{i=1}^{\kappa} \frac{1}{(1-p_i)^2}\right)$$

When $\log_2 \frac{d'}{u}$ is an integer, $\kappa = \log_2 \frac{d'}{u}$. Then the number of tests is:

$$m = O\left(\left(\frac{du}{2}\right)^{\frac{1}{2} \log_2 \frac{d'}{u}} \cdot \prod_{i=1}^{\log_2 \frac{d'}{u}} \ln \left(\frac{n}{d_i'} \cdot 2^i\right) \cdot \prod_{i=1}^{\log_2 \frac{d'}{u}} \frac{1}{(1-p_i)^2}\right).$$

This completes the proof.