Efficient Document Exchange and Error Correcting Codes with Asymmetric Information*

Kuan Cheng† Xin Li‡

Abstract

We study two fundamental problems in communication, Document Exchange (DE) and Error Correcting Code (ECC). In the first problem, two parties hold two strings, and one party tries to learn the other party's string through communication. In the second problem, one party tries to send a message to another party through a noisy channel, by adding some redundant information to protect the message. Two important goals in both problems are to minimize the communication complexity or redundancy, and to design efficient protocols or codes.

Both problems have been studied extensively. In this paper we study whether asymmetric partial information can help in these two problems. We focus on the case of Hamming distance/errors, and the asymmetric partial information is modeled by one party having a vector of disjoint subsets $\mathbf{S} = (S_1, \cdots, S_t)$ of indices and a vector of integers $\mathbf{k} = (k_1, \cdots, k_t)$, such that in each S_i the Hamming distance/errors is at most k_i. To our knowledge, no previous work has studied this problem systematically. We establish both lower bounds and upper bounds in this model, and provide efficient randomized constructions that achieve a $\min\{O(t^2), O\left((\log\log n)^2\right)\}$ factor within the optimum, with almost linear running time.

We further show a connection between the above document exchange problem and the problem of document exchange under *edit distance*, and use our techniques to give an efficient randomized protocol with optimal communication complexity and *exponentially* small error for the latter. This improves the previous result by Haeupler [20] (FOCS'19), which has polynomially large error; and that by Belazzougui and Zhang [8] (FOCS'16), which is only optimal for a limited range of parameters.

Our techniques are based on a generalization of the celebrated expander codes by Sipser and Spielman [36], which may be of independent interests.

1 Introduction

Document exchange is a combinatorial version of the famous Slepian-Wolf problem [37], which is a fundamental problem in communication and coding theory dating back to 1973. It was then studied by Orlitsky [32] and subsequently named and also studied by Cormode et. al. [15]. Here, two parties Alice and Bob each holds a string (document) x and y, and the goal is for one party to learn the other party's string with the least amount of communication possible. For simplicity, let us assume that both x and y have n bits. If x and y can be arbitrary strings, then it is clear that in the worst case the communication needs at least n bits, i.e., sending one party's string to the other party. However, in practice this is often not the case, and x and y can actually be close in some sense. For example, Alice and Bob may be two uses holding different versions of some original document, where x and y are obtained after some edits of a string z. If the number of edits is limited, then it is possible for one party to learn the other party's string with significantly less amount of communication. In this paper, we focus on the case where the strings have binary alphabet.

More generally and formally, the document exchange problem can be described as follows. Alice and Bob each has an n-bit string x and y, and the distance between x and y, $D(x, y)$ is upper bounded by some number k. Here the distance D can be any measure of interests. Now, the first goal here is to minimize the communication complexity as a function of n and k. In addition, it is also an important goal to keep the protocol *efficient*, i.e., we would like the communication protocol to run in polynomial time of n.

There has been a lot of work on the document exchange problem [32, 6, 7, 1, 15, 30, 38, 25, 26, 8, 12, 20, 13]. While Orlitsky [32] established some upper and lower bounds on the communication complexity of general "balanced" measures $D(x, y)$, as well as

*A full version of this paper appears in [14]

†Center on Frontiers of Computing Studies, Peking University. ckkcdh@pku.edu.cn. Supported in part by a start-up funding of Peking University CFCS, a Simons Investigator Award (#409864, David Zuckerman) and NSF Award CCF-1617713.

‡Department of Computer Science, Johns Hopkins University. lixints@cs.jhu.edu. Supported by NSF Award CCF-1617713 and NSF CAREER Award CCF-1845349.

exponential time protocols that can achieve the optimal communication, efficient protocols in subsequent works have been mostly focusing on the two natural cases where $D(x, y)$ is either the Hamming distance or the edit distance. In the former, the distance is measured by how many bits in x and y are different at the corresponding locations, while in the latter the distance $\mathsf{ED}(x, y)$ is measured by the minimum number of insertions, deletions, and substitutions to transform one string into another. Both distances are metrics, and edit distance strictly generalizes Hamming distance.

For both Hamming distance and edit distance, it is known that if $D(x, y) \leq k$, then the optimal communication complexity in the document exchange problem is $\Theta(k \log(n/k))$, and this can be achieved by a deterministic one-round protocol running in exponential time. The situation of efficient protocols however is different for these two measures. For Hamming distance, we have many efficient, deterministic one-round protocols with optimal communication complexity $\Theta(k \log(n/k))$, using checksum decoding of error-correcting codes such as Algebraic Geometry codes [24], BCH codes [9, 23], etc. For edit distance, except for the exponential time deterministic one-round protocol in [32] which achieves optimal communication complexity, for a long time only efficient randomized one round protocols with sub-optimal communication complexity are known. These include the work of Irmak et al. [25] with communication complexity $O(k \log(\frac{n}{k}) \log n)$, the work of Jowhari [26] with communication complexity $O(k \log^2 n \log^* n)$, the work of Chakraborty et al. [11] with communication complexity $O(k^2 \log n)$, and the work of Belazzougui and Zhang [8] with communication complexity $O(k(\log^2 k + \log n))$. In particular, the protocol in [8] has asymptotically optimal communication complexity for $k = 2^{O(\sqrt{\log n})}$, with success probability $1 - 1/\mathsf{poly}(k \log n)$.

In 2018, Cheng et. al. [12], and Haeupler [20] independently gave an efficient, deterministic one-round protocol with communication complexity $O(k \log^2(n/k))$. Finally, Haeupler [20] gave the first efficient randomized one-round protocol with optimal communication complexity $O(k \log(n/k))$. However, his protocol only succeeds with probability $1 - 1/\mathsf{poly}(n)$.

Document exchange is closely related to the (even more) fundamental problem of error correcting codes. The goal of an error correcting code is to ensure that one party can successfully send information to another party, despite errors caused by the communication channel. In this setting, the first party (Alice) runs an encoding algorithm that turns a message of m bits into a codeword of n bits, and sends the codeword to the second party (Bob) through a channel. Bob then tries to recover the message by running a decoding algorithm. Similar to document exchange, there are also two important goals here. First, one wants to keep $n - m$ (the redundancy of the codeword) to be as small as possible, or alternatively, to keep m (the message length) to be as large as possible. Second, one needs both the encoding and decoding to be efficient, i.e., run in polynomial time of m.

There has been extensive study on error correcting codes, which we will not be able to completely survey here. Again, the channel error can have several different models, and the most studied are Hamming errors and edit errors. For both cases, assuming k is an upper bound on the number of errors, then it is known that the optimal message length one can achieve (with possibly exponential time encoding/decoding) is $m = n - \Theta(k \log(n/k))$. For Hamming errors, again we have efficient constructions matching this bound, based on Algebraic Geometry codes [24]. For edit errors the constructions are far behind, and for a long time we only have asymptotically optimal constructions for the two extreme cases of $k = 1$ [29] and $k = \alpha n$ for some small constant $\alpha > 0$ [33]. A recent line of works [18, 17, 10, 21, 22, 12, 20] achieved significant progress on this problem. In particular, Cheng et. al. [12], and Haeupler [20] independently gave an efficient code with $m = n - O(k \log^2(n/k))$. Cheng et. al. [12] further gave an efficient code with $m = n - O(k \log n)$, which is optimal for $k \leq n^{1-\alpha}$ where $\alpha > 0$ is any constant.

The connection between document exchange and error correcting codes is demonstrated by the notion of systematic error correcting codes. These are codes where a codeword is simply the message followed by some redundant information called the *checksum*. Given such a code, the checksum can be used as the information sent in a document exchange protocol. Conversely, given a one round document exchange protocol, one can use a standard error correcting code on the information sent and use it as the checksum in a systematic error correcting code.

In all previous works, Alice and Bob have symmetric information—they both know that their string is within distance $D(x, y) \leq k$ to the other party's string, or the total number of errors in the received codeword is at most k. However, in many practical situations, each party may have some additional partial information that is not known to the other party. For example, in document exchange, if Bob has made edits in some specific parts of the original document, then even without carefully tracking the edits, Bob has some partial information of where the differences can happen. This information is not necessarily known to Alice. In another situation, suppose Alice sends a long string to

Bob by Internet routing, then this string may be broken into several parts and transmitted to Bob through different channels. These channels may have different behavior and introduce different numbers of errors. While it is reasonable that both parties know the parameters of all channels, due to the routing process Alice may not know which channels her parts are sent through. On the other hand, Bob can learn these information by observing the received parts. Thus Bob will have some partial information about the numbers of errors in specific parts of the received string, which is not known to Alice. The fist example applies to document exchange and the second example applies to error correcting codes. One can now ask the following natural question, which is the focus of this paper.

Question: *Can we use these asymmetric information to reduce the communication complexity in document exchange or the redundancy in error correcting codes, while still designing efficient protocols or codes?*

Towards answering this question, we first formally define our model.

1.1 The Model of Asymmetric Information In this paper we focus on Hamming distance/Hamming errors in the model of asymmetric information. To model the asymmetric information, we assume that one party has some additional information of where the differences/errors can happen. More formally, we use a vector of disjoint subsets $\mathbf{S} = (S_1, \cdots, S_t)$ to indicate the positions where the differences/errors can happen, and a vector of integers $\mathbf{k} = (k_1, \cdots, k_t)$ to indicate the upper bounds on the numbers of differences/errors in each set S_i. For each S_i, let s_i denote the size of S_i, i.e., $s_i = |S_i|$. We also use $\mathbf{s}$ to indicate the vector $\mathbf{s} = (s_1, \cdots, s_t)$. We assume the parameters $(\mathbf{s}, \mathbf{k}, t)$ are known to both parties, and that (without loss of generality) $k_1 \geq k_2 \geq \cdots \geq k_t$.

Definition 1.1. *($(\mathbf{s}, \mathbf{k}, t)$ Asymmetric Document Exchange) There are two parties Alice and Bob. Alice has a string $x \in \{0,1\}^n$ and Bob has a string $y \in \{0,1\}^n$. Both parties know $(\mathbf{s}, \mathbf{k}, t)$. In addition, Bob knows a vector of disjoint subsets $\mathbf{S} = (S_1, \cdots, S_t)$ where $\forall i, S_i \subseteq [n]$ and $|S_i| = s_i$. That is, within each set S_i, the Hamming distance between x and y is at most k_i. One party tries to learn the string of the other party.*

Definition 1.2. *($(\mathbf{s}, \mathbf{k}, t)$ Asymmetric Error Correcting Code) There are two parties Alice and Bob. Both parties know $(\mathbf{s}, \mathbf{k}, t)$. Alice encodes a message of m bits into a codeword of n bits, using a function $\mathsf{Enc} : \{0,1\}^m \to \{0,1\}^n$ and sends it to Bob. Bob knows a vector of disjoint subsets $\mathbf{S} = (S_1, \cdots, S_t)$ where $\forall i, S_i \subseteq [n]$ and $|S_i| = s_i$. That is, within each set S_i, there are at most k_i Hamming errors in the received codeword. Bob uses a function $\mathsf{Dec} : \{0,1\}^n \to \{0,1\}^m$ to recover the message.*

We require the protocol or code to succeed for *every* possible vector of disjoint subsets $\mathbf{S} = (S_1, \cdots, S_t)$ with $|S_i| = s_i, \forall i$, and for *every* possible distance/error pattern that is consistent with $\mathbf{S} = (S_1, \cdots, S_t)$ and $\mathbf{k} = (k_1, \cdots, k_t)$.

We consider both deterministic and randomized protocols/codes. In the case of randomized solutions, we assume that the two parties have shared randomness, as is standard in all previous works. In the case of error correcting codes, we further assume that the channel errors do not depend on the shared randomness.

Our model is quite general in capturing asymmetric information. A naive solution is to simply ignore the extra information, and apply a document exchange protocol or error correcting code for $k = \sum_{i=1}^{t} k_i$ Hamming distance or Hamming errors. However, our goal here is to see if the extra information can be used to design better protocols or codes. Another natural strategy for the document exchange problem, is for Bob to first send the descriptions of $\mathbf{S}$ to Alice, and they can then run a protocol on each set S_i. However, this strategy can result in a significant amount of communication, e.g., $\sum_{i=1}^{t} s_i \log n$, which can be even larger than n. In some special situations, a set S_i may be a continuous block in the string, and it suffices to just send the starting and ending index, using $2 \log n$ bits. If all sets S_i are of this form, then the total number of bits required is $2t \log n$. Even this number can be large when the number of sets t is large. We also stress that in our model and all results, each set S_i does not need to be a continuous block. A final simple strategy is to try to form a large continuous block which includes several S_i's, but this can increase the size of the sets significantly and thus also results in a penalty on the communication complexity.

Remark 1.3. *In the asymmetric document exchange, it may seem unreasonable to assume that Alice knows the vectors $\mathbf{s}, \mathbf{k}$. However, this is without loss of generality up to a small loss in communication complexity and communication rounds. Basically, Bob can first send these two vectors to Alice. This only takes one round and the number of bits sent by Bob is $O(\sum_{i=1}^{t}(\log k_i + \log s_i))$, while the number of bits needed to distinguish all possible error patters is at least $\sum_{i=1}^{t} \log \binom{s_i}{k_i}$. The former is always within a constant factor to (and in most cases smaller than) the latter.*

Related previous works. While document exchange and error correcting codes with asymmetric information are natural questions, to our knowledge they have not been studied systematically. The only previous work we found is the work of Belazzougui and Zhang [8], which studies a special case of our model with $t = 1$, i.e., Bob's extra information only has one subset S with $|S| = s$. They use entirely different techniques to give a document exchange protocol with sub-optimal communication complexity $O(k(\log s + \log(1/\varepsilon)))$, where Bob can learn Alice's string with success probability $1 - \varepsilon$.

However, there are a large body of works on a related topic [3, 28, 27, 39, 2, 5], which study the problem of source coding/data compression with asymmetric information. In this setting, the decoder has some prior distribution μ not known to the encoder, and the encoder tries to send a set of items drawn independently from the distribution to the decoder, using the smallest number of bits as possible. The problem we study here, on the other hand, focuses on error correction. While there are similarities between these two problems, they are also fundamentally different. For example, all the efficient algorithms in these prior works run in time polynomial in the size of the support μ. This is prohibitive for our purpose since this number is already exponentially large.

We note that source coding and error correction are the two most important applications of information theory. Thus given the abundant works on source coding/data compression with asymmetric information, we believe a systematic study of document exchange and error correcting codes with asymmetric information is also an important direction.

1.2 Our Results We provide both lower bounds and upper bounds for document exchange and error correcting codes with asymmetric information. To simplify the presentation, we first define some quantities. Given two vectors $\mathbf{s} = (s_1, \cdots, s_t)$ and $\mathbf{k} = (k_1, \cdots, k_t)$, we define $\mathsf{H}(\mathbf{s}, \mathbf{k}) = \log\left(\prod_{i=1}^{t}\left(\sum_{j=0}^{k_i}\binom{s_i}{j}\right)\right) = \sum_{i=1}^{t}\log\left(\sum_{j=0}^{k_i}\binom{s_i}{j}\right)$. Similarly, for two integers s and k with $s \geq k$, we define $\mathsf{H}(s, k) = \log\left(\sum_{j=0}^{k}\binom{s}{j}\right)$.

Note that if $\forall i, s_i \geq 2k_i$ and $s \geq 2k$, then $\mathsf{H}(\mathbf{s}, \mathbf{k}) = \Theta(\sum_{i=1}^{t} k_i \log(s_i/k_i))$ and $\mathsf{H}(s, k) = \Theta(k \log(s/k))$. Recall that $k = \sum_{i=1}^{t} k_i$ and $s = \sum_{i=1}^{t} s_i \leq n$, hence $\mathsf{H}(\mathbf{s}, \mathbf{k}) \leq \mathsf{H}(n, k)$. We have the following theorem.

THEOREM 1.1. *In an $(\mathbf{s}, \mathbf{k}, t)$ asymmetric DE problem, we have*

- *Suppose Alice learns Bob's string, then any deterministic protocol has communication complexity at least $\mathsf{H}(n, k)$, and any randomized protocol with success probability $\geq 1/2$ has communication complexity at least $\mathsf{H}(n, k) - 1$.*
- *Suppose Bob learns Alice's string, then any randomized protocol with success probability $\geq 1/2$ has communication complexity at least $\mathsf{H}(\mathbf{s}, \mathbf{k}) - 1$. Furthermore if $\forall i, s_i \geq 2k_i$, then any one round deterministic protocol has communication complexity at least $\mathsf{H}(n, k)$.*

This theorem tells us the following important things: First, Bob's extra information is only useful for him to learn Alice's string, but not useful in the other direction. Second, in the case of a one round protocol for Bob to learn Alice's string, for a wide range of parameters (i.e., when $\forall i, s_i \geq 2k_i$), Bob's extra information is only useful in randomized protocols.

For upper bounds, we note that there are efficient deterministic protocols to meet the bound $\mathsf{H}(n, k)$, based on algebraic geometry codes. To meet the bound $\mathsf{H}(\mathbf{s}, \mathbf{k})$, there is also a simple one round randomized protocol: Alice hashes her string x using a random hash function, and Bob enumerates all possible strings to find the one with the correct hash value. It's easy to see that this protocol succeeds if there is no hash collision, which happens with high probability if the hash function outputs some $O(\mathsf{H}(\mathbf{s}, \mathbf{k}))$ bits. However, this protocol runs in exponential time, and our main result is an *efficient* protocol that gets close to this bound.

To state our main theorem, we define another quantity $\chi(s, k, t) \in \mathsf{N}$: first partition the interval $[2, n]$ into disjoint subintervals $\{I_j = [2^{10^{j-1}}, 2^{10^j})\}$, starting from $j = 1$. Then, for every $i \in [t]$, put s_i/k_i into the corresponding subinterval. $\chi(s, k, t)$ is defined to be the number of subintervals I_j which contain at least one s_i/k_i. We now have the following theorem.

THEOREM 1.2. *In an $(\mathbf{s}, \mathbf{k}, t)$ asymmetric DE problem, suppose that $\forall i, s_i \geq 2k_i$. There is an efficient randomized one round protocol for Bob to learn Alice's string, with communication complexity $O(\chi(s, k, t)^2 \mathsf{H}(\mathbf{s}, \mathbf{k}))$ and error probability $2^{-\Omega(k_t)} + \frac{1}{\mathsf{poly}(s)}$. The protocol runs in time $\widetilde{O}(n)$.*

Note that $\chi(s, k, t) \leq t$ and $\chi(s, k, t) \leq \log\log n$, so the above theorem immediately gives the following two corollaries.

COROLLARY 1.1. *In an $(\mathbf{s}, \mathbf{k}, t)$ asymmetric DE problem, suppose that $\forall i, s_i \geq 2k_i$. There is an efficient randomized one round protocol for Bob to learn Alice's string, with communication complexity $O(t^2 \mathsf{H}(\mathbf{s}, \mathbf{k}))$ and*

error probability $2^{-\Omega(k_t)} + \frac{1}{\mathsf{poly}(s)}$. *The protocol runs in time* $\widetilde{O}(n)$.

COROLLARY 1.2. *In an* $(\mathbf{s}, \mathbf{k}, t)$ *asymmetric DE problem, suppose that* $\forall i, s_i \geq 2k_i$. *There is an efficient randomized one round protocol for Bob to learn Alice's string, with communication complexity* $O((\log\log n)^2 \mathsf{H}(\mathbf{s}, \mathbf{k}))$ *and error probability* $2^{-\Omega(k_t)} + \frac{1}{\mathsf{poly}(s)}$. *The protocol runs in time* $\widetilde{O}(n)$.

In particular, Corollary 1.1 implies that if t is a constant, then we have a one round protocol with asymptotically optimal communication complexity, while Corollary 1.2 gives a one round protocol with communication complexity optimal up to an additional $(\log\log n)^2$ factor. Both protocols run in near linear time. We also note that the simple strategy of ignoring the extra information can result in communication complexity $\Omega(\mathsf{H}(\mathbf{s}, \mathbf{k}) \log n)$ in the worst case.

Similarly, we have both lower bounds and upper bounds for error correcting codes with asymmetric information. The first theorem shows that such information is only useful for a randomized code.

THEOREM 1.3. *In an* $(\mathbf{s}, \mathbf{k}, t)$ *asymmetric ECC problem, if* $\forall i, s_i = |S_i| \geq 2k_i$, *then any deterministic code must have distance at least* $2k + 1$. *In particular, this means* $m \leq n - \mathsf{H}(n, k)$. *Furthermore, any randomized code with success probability* $\geq 1/2$ *must have message length* $m \leq n - \mathsf{H}(\mathbf{s}, \mathbf{k}) + 1$.

Again, a code with randomized encoding and exponential time deterministic decoding can achieve message length $m = n - O(\mathsf{H}(\mathbf{s}, \mathbf{k}))$. We design an efficient code that comes close to this.

THEOREM 1.4. *In an* $(\mathbf{s}, \mathbf{k}, t)$ *asymmetric ECC problem, suppose* $\forall i, s_i \geq 2k_i$. *There is an efficient code with randomized encoding and deterministic decoding, which has message length* $m = n - O(\chi(s, k, t)^2 \mathsf{H}(\mathbf{s}, \mathbf{k}))$ *and error probability* $2^{-\Omega(k_t)} + \frac{1}{\mathsf{poly}(s)}$. *In particular, the message length can be* $\max\{n - O(t^2 \mathsf{H}(\mathbf{s}, \mathbf{k})), n - O((\log\log n)^2 \mathsf{H}(\mathbf{s}, \mathbf{k}))\}$, *and the running time is* $\widetilde{O}(n)$.

Next we show that we can design efficient document exchange protocols with asymptotically optimal communication complexity in a special case, roughly when $\mathbf{s}, \mathbf{k}$ are geometric progressions.

THEOREM 1.5. *There is an efficient randomized one-round protocol for every* $(\mathbf{s}, \mathbf{k}, t)$ *asymmetric DE problem, where* $s_i = k2^{\Theta(i)}, k_i = \max\{k/2^{\Theta(i)}, \Theta(\frac{k}{t^2 \log \frac{n}{k}})\} \leq s_i/40$. *The communication complexity is* $O(k)$ *and the error probability is* $2^{-\Omega(k/\log \frac{n}{k})}$.

We show that the problem of document exchange under edit distance can be reduced to the special case above, and thus we obtain the following theorem.

THEOREM 1.6. *There is an efficient randomized one-round protocol for the DE problem with edit distance at most* k. *The communication complexity is* $O(k \log \frac{n}{k})$ *and the error probability is* $\min\{2^{-\Theta(k/\log^3 \frac{n}{k})}, 1/\mathsf{poly}(n)\}$.

We also have both lower bounds and upper bounds for document exchange where both parties have some asymmetric partial information, represented as a vector of disjoint subsets. For the clarity of presentation we omit the results here, and refer the reader to Section **??** for details.

1.3 Technique Overview Our lower bounds follow from relatively simple information theoretic arguments, so here we only provide an informal outline of our protocols. We start with the asymmetric document exchange for Hamming distance. Recall that the asymmetric information is in the form of $\mathbf{S} = (S_1, \cdots, S_t)$ and $\mathbf{k} = (k_1, \cdots, k_t)$, where $\forall i, |S_i| = s_i$ and the Hamming distance within S_i is at most k_i. We assume $\forall i, s_i \geq 2k_i$, and without loss of generality that $k_1 \geq k_2 \cdots \geq k_t$.

The protocol for one set. Our starting point is the simplest case where $t = 1$, i.e. there is only one set S of size s and the Hamming distance in S is at most k. In this case our goal is to give an efficient one round protocol with communication complexity $O(k \log \frac{s}{k})$. If $s = n$ then this can be achieved by using a systematic algebraic geometry code or an expander code [35]. We will use the latter and we briefly review the application of expander codes in document exchange.

To run the protocol, the two parties choose a bipartite expander graph $G : [n] \times [d] \to [m]$. Alice associates her string x with the n vertices on the left, and computes a string z of length m as follows: For every $i \in [m]$, let $z_i = \bigoplus_{j \in \Gamma^{-1}(i)} x_j$, where $\Gamma^{-1}(i)$ is the set of neighbors of the right vertex i in the expander. The string z consists of a sequence of parity checks of x, and is then sent to Bob.

To recover x, Bob starts out with $\tilde{x} = y$ as his current version of x, and maintains another string $z' \in \{0,1\}^m$ using the same approach as above, except replacing the string x by $\tilde{x}$, i.e., z' consists of a sequence of parity checks of $\tilde{x}$. z and z' will differ in several coordinates, and Bob will gradually modify $\tilde{x}$ into x by flipping some bits in $\tilde{x}$ according to the parity checks. This process is known as belief propagation, and works as follows. Bob keeps finding a bit in $\tilde{x}$ such that by flipping this bit, the Hamming distance between z' and z decreases by at least one. Bob flips this bit and updates $\tilde{x}$ and z'

correspondingly. Bob stops when $z' = z$, at which point $x = \tilde{x}$ and he has successfully recovered x.

For the analysis, we use the set $R \subseteq [n]$ to denote the coordinates where x and $\tilde{x}$ are different. We say the i'th parity check bit is satisfied if $z_i = z'_i$, and unsatisfied otherwise. Let the number of satisfied and unsatisfied checks in $\Gamma(R)$ (the neighbors of R) be s and u. Assume the graph has good expansion, i.e. $|\Gamma(R)| = \mathsf{s} + \mathsf{u} \geq 0.9d|R|$, and note that in $\Gamma(R)$, each satisfied check has at least two neighbors in R. Thus $2\mathsf{s} + \mathsf{u} \leq d|R|$. By the two inequalities, we deduce $\mathsf{u} \geq 0.8d|R|$ and thus at least one left vertex has more unsatisfied parity checks as neighbors than satisfied parity checks, and Bob can flip this bit. The analysis holds as long as the expansion of the set R is guaranteed. Note that the number of unsatisfied checks is strictly decreasing in the process, thus $|R|$ can never be more than $1.25k$, since otherwise this will induce more than dk unsatisfied checks, but at the beginning there are at most dk unsatisfied checks. Therefore, we only need to guarantee the expansion of all $R \subseteq [n]$ with $|R| \leq 1.25k$, and a random graph with $m = O(k \log \frac{n}{k})$ and $d = O(\log \frac{n}{k})$ satisfies this property with high probability.

Going back to the case where $s < n$, the first issue is that we can't afford to use an expander which has good expansion for all subsets R as before, since this will make $m = \Omega(k \log \frac{n}{k})$. To fix this, we instead just require the expansion to hold for all subsets $R \subseteq S$ with $|R| \leq 1.25k$. Now, a random graph with $m = O(k \log \frac{s}{k})$ and $d = O(\log \frac{s}{k})$ satisfies this property with high probability, and both parties can generate the same expander by using the shared randomness. Similarly, when recovering x Bob will always look for a bit in S to flip. The analysis is now similar to the standard case and this gives the protocol for the case of $t = 1$.

The protocol for two sets. We now consider the case with $t > 1$. Our goal is to design an efficient one round protocol with communication complexity close to $\mathsf{H}(\mathbf{s}, \mathbf{k})$.

The first idea may be to take the union of all $S_i, i \in [t]$ as one set S, and the Hamming distance in S is at most $k = \sum_{i \in [t]} k_i$. Now we can use the protocol for $t = 1$ described before. However, in this case the communication complexity will be $O(\mathsf{H}(s, k))$, which may not be close to $\mathsf{H}(\mathbf{s}, \mathbf{k})$. For example, consider the case where $t = 2$, $k_1 = n^{0.1}, s_1 = 10n^{0.1}$, $k_2 = 10, s_2 = 0.1n$. A direct computation indicates $\mathsf{H}(s, k) = \Omega(\mathsf{H}(\mathbf{s}, \mathbf{k}) \log n) = \omega(\mathsf{H}(\mathbf{s}, \mathbf{k}))$. It also appears hard to improve this if we just use a single expander graph, since the decoding requires good expansion for all possible subsets of errors during the belief propagation, which can potentially be all possible subsets of size $\Omega(k)$. This forces the right hand size of the graph to be $\mathsf{H}(s, k)$.

To overcome this difficulty, our idea is to use more than one expander codes. Towards this, our main observation is that, the issue with the above example is due to the following fact: for some $i \in [t]$, k_i is large while s_i is small, but for some other i, k_i is small while s_i is large. Indeed, in the case of $t = 2$, there are two good situations where $\mathsf{H}(s, k) = O(\mathsf{H}(\mathbf{s}, \mathbf{k}))$:

1. k_1 and k_2 are roughly the same, i.e., $k_1 = \Theta(k_2)$. In this case we have $\mathsf{H}(s, k) = \Theta((k_1 + k_2) \log \frac{s_1 + s_2}{k_1 + k_2}) = \Theta(k_1 \log \frac{s_1}{k_1} + k_2 \log \frac{s_2}{k_2}) = \Theta(\mathsf{H}(\mathbf{s}, \mathbf{k}))$.
2. $\log \frac{s_1}{k_1}$ and $\log \frac{s_2}{k_2}$ are roughly the same, i.e., $\log \frac{s_1}{k_1} = \Theta(\log \frac{s_2}{k_2})$. In this case we also have $\mathsf{H}(s, k) = \Theta(\mathsf{H}(\mathbf{s}, \mathbf{k}))$.

Our protocol will exploit both of these good cases. We first illustrate this with a protocol for the case of $t = 2$. Our idea is to reduce the number k_1 (recall that $k_1 \geq k_2$) to be roughly the same as k_2 (which is unnecessary if k_1 and k_2 are already roughly the same at the beginning). In other words, we will first reduce the Hamming distance in S_1 from k_1 to at most ck_2, if $k_1 > ck_2$ for some constant $c > 1$. It is not immediately clear why this is feasible, since Alice does not know the subset S_1. Additionally, we need to make sure the communication complexity of this step is not too large.

We achieve this by using an expander code based on a bipartite expander $G : [n] \times [d] \to [m]$ such that for all sets $R \subseteq S_1$ with $|R| \in [ck_2, 1.6k_1]$, the set R has good expansion, i.e., $|\Gamma(R)| \geq 0.9d|R|$. The expander is again generated by shared randomness, and we show that we can choose $d = O(\log \frac{s_1}{k_1}), m = O(k_1 \log \frac{s_1}{k_1})$ and the graph satisfies the property with high probability. Alice will again compute the parity checks z and send it to Bob.

Now Bob will apply the same method as before: start with $\tilde{x} = y$ and keep finding a bit in S_1 with more unsatisfied parity checks as neighbors than satisfied parity checks. Bob flips this bit and continues doing this until no such bit can be found. Since the number of unsatisfied parity checks keeps decreasing, the process will end in a finite number of steps. We claim that when it ends, the Hamming distance in S_1 is at most ck_2. This effectively reduces the Hamming distance in S_1.

The main issue in the analysis here is that the different bits between x and y are not entirely in S_1, and this may cause problems in belief propagation. However, our observation is that when k_1 is much larger than k_2, the effect of k_2 can mostly be ignored. More specifically, let R be the set of left vertices which correspond to the

different bits between x and y in S_1, and R_2 be the set of left vertices which correspond to the different bits in S_2. Thus $|R_2| \leq k_2$. Let the number of satisfied and unsatisfied checks in $\Gamma(R_1)$ be s and u. As long as $|R| \in [ck_2, 1.6k_1]$, we have $|\Gamma(R)| = \mathsf{s}+\mathsf{u} \geq 0.9d|R|$, and $2\mathsf{s}+\mathsf{u} \leq d|R| + d|R_2| \leq (1+\frac{1}{c})|R|$. Combining these inequalities, we can still deduce $u \geq 0.7d|R|$, by setting $c = 10$. Hence there must exist a bit in S_1 to flip. Since the number of unsatisfied checks decreases strictly, the size $|R|$ in the process can never be larger than $1.6k_1$. This is because otherwise there will be at least $1.12dk_1$ unsatisfied checks, while at the beginning there are only at most $(1+1/c)dk_1 = 1.1dk_1$ unsatisfied checks. Thus when this process stops, we must have $|R| \leq ck_2$. At this point, we can use the protocol for one set together with another expander graph to finish the job, by considering the set $S = S_1 \cup S_2$ which has Hamming distance at most $(c+1)k_2$. The total communication complexity is $O(k_1 \log \frac{s_1}{k_1}) + O(k_2 \log \frac{s_1+s_2}{k_2}) = O(\mathsf{H}(\mathbf{s}, \mathbf{k}))$.

The protocol for arbitrary t. We now generalize the above protocol to arbitrary t. Recall that $k_1 \geq k_2 \geq \cdots \geq k_t$. Our idea is to use the above protocol of reducing Hamming distance repeatedly, while going through the index from 1 to t. More formally, we use i' to denote the current index and k' to denote an upper bound of the Hamming distance in $\cup_{j\in[i']} S_j$ after possible steps of reducing distance. We start with $i' = 0, k' = 0$ and repeat the following: find the first index $i > i'$ s.t. the current Hamming distance in $\cup_{j\in[i]} S_j$ is much larger than the Hamming distance in $\cup_{j=i+1}^{t} S_j$, i.e.,

$$k' + \sum_{j=i'+1}^{i} k_j > c \sum_{j=i+1}^{t} k_j = k''. \tag{1.1}$$

Then we reduce the Hamming distance in $\cup_{j\in[i]} S_i$ to at most k'' by using the two set protocol described before, regarding $\cup_{j\in[i]} S_j$ as one set and $\cup_{j=i+1}^{t} S_j$ as the other set. We now update $k' = k'', i' = i$ and continue the process. Finally, the Hamming distance in $S = \cup_{j\in[t]} S_i$ will be reduced to at most $(c+1)k_t$, and we apply the one set protocol for S to finish the job.

The correctness follows from the correctness of the one set protocol and the two set protocol. The main thing left is to bound the communication complexity. Note that except the first iteration, in each subsequent iteration i' will be updated to $i'+1$. Thus the number of bits Alice sends in this step is $m = O\left((k'+k_i)\log \frac{\sum_{j\in[i]} s_j}{k'+k_i}\right)$. We show that this is always $O(t\mathsf{H}(\mathbf{s}, \mathbf{k}))$ by using the bound on k', the fact that $k_1 \geq k_2 \geq \cdots \geq k_t$, and $k_i \leq s_i/2, \forall i \in [t]$. Thus the total communication complexity is $O(t^2\mathsf{H}(\mathbf{s}, \mathbf{k}))$. Note that this is a one round protocol since only Alice sends out information.

Finally, we can get further improvement by grouping some sets together. Specifically, we divide the interval $[2, n]$ into disjoint subintervals $I_j = [2^{10^{j-1}}, 2^{10^j}), j = 1, \ldots, O(\log\log n)$ and put each subset S_i into one interval according to the number s_i/k_i. Whenever two subsets S_i and S_j are in the same interval, we have $\log(s_i/k_i) = \Theta(\log(s_j/k_j))$ and thus we can consider $S_i \cup S_j$ as one set with Hamming distance $k_i + k_j$, without changing the communication complexity much. Now, taking the union of all subsets in the same interval to be one subset reduces the number of subsets to $\chi(s, k, t)$, and applying our protocol results in communication complexity $O(\chi(s, k, t)^2\mathsf{H}(\mathbf{s}, \mathbf{k}))$.

ECC with asymmetric information. The protocol for document exchange can be used to construct an error correcting code. We do this by first estimating the length of the redundant information. Let m_0 be the communication complexity of the $(\mathbf{s}, \mathbf{k}, t)$ DE protocol for message length n. We choose an asymptotically good code C_0 with message length m_0 and codeword length n_0, which corrects k errors. The actual message length of our code will be $n - n_0$. On input message x, we run Alice's DE protocol on $x \circ \mathbf{0}$ where $\mathbf{0} = 0^{n_0}$ to get $z \in \{0,1\}^{m_0}$. Then we encode z by C_0 and the final codeword is $x \circ C_0(z)$. To decode, one first recovers z by running the decoding algorithm of C_0 on the part $C_0(z)$. Then we run Bob's DE protocol using z, and by replacing the $C_0(z)$ part with 0^{n_0}. The correctness follows from the code C_0 and the DE protocol.

1.3.1 Document exchange under edit distance

We now describe our protocol for document exchange under edit distance, and show a connection to the problem of document exchange under Hamming distance with asymmetric information.

On a high level, our protocol follows the leveled structure used in several previous works [25, 12, 20]. The protocol proceeds in $L = O(\log(\frac{n}{k}))$ levels where in each level, Alice sends a sketch of her string x with $O(k)$ bits to Bob. Bob then uses all the sketches and his string y to recover x.

On Alice's side, in the first level she divides her string into $\Theta(k)$ blocks where each block has size $O(\frac{n}{k})$. In each subsequent level, every block from the previous level is divided evenly into two blocks, and this ends when the block size becomes $O(\log \frac{n}{k})$, which takes $O(\log(\frac{n}{k}))$ levels. In each level, Alice applies a different random hash function to every block using the shared randomness, and computes a sketch based on the hash values. On Bob's side, his recovering process also proceeds in L levels, where in each level Bob maintains a string $\tilde{x}$ which is Bob's current version of Alice's string x. Specifically,

in each level Bob also applies the same hash functions to the blocks of $\tilde{x}$ to get the hash values, then he uses this level's sketch to recover the correct hash values of Alice's bocks. Bob will then find the blocks in $\tilde{x}$ which have inconsistent hash values with Alice's blocks, and update these blocks using his string y by computing a non overlapping matching between y's blocks and the corresponding hash values. An important property of the protocol is that in each level, the number of different blocks between x and $\tilde{x}$ is always bounded by $O(k)$ with high probability. This ensures that Alice can send a short sketch to Bob for him to recover the correct hash values of all blocks.

To ensure that Alice's sketch in each level has length $O(k)$, there are several non trivial issues. First, every hash function needs to have only $O(1)$ bits of output, as in [20]. Second, even so, the general task of recovering s hash values with $O(k)$ errors needs to use a sketch of size at least $\log \binom{s}{k} = \Omega(k \log \frac{s}{k})$, where s is the number of blocks in the current level. This can be as large as $\Omega(k \log \frac{n}{k})$ when s becomes $n^{\Omega(1)}$, and thus will be problematic. To fix this issue, [20] uses a more careful analysis called "t-witness" to show that in each level, the total number of possible error patterns is $2^{O(k)}$ with high probability, instead of $\binom{s}{k}$. Thus, in theory one can simply use another random hash function with $O(k)$ bits of output to distinguish all error patterns, and this brings the sketch size back to $O(k)$. However, simply doing this will result in an exponential running time since it involves exhaustive search. Thus, [20] needs to first randomly partition the blocks into bins, such that with high probability each bin has $O(\log n)$ hash errors. The exhaustive search in each bin now takes $\mathsf{poly}(n)$ time. Unfortunately, this also increases the error probability from $2^{-\Omega(k)}$ to $1/\mathsf{poly}(n)$.

In our protocol, we instead replace the approach of random partitioning and exhaustive search in [20] by a direct efficient approach, thus improving the error probability to be exponentially small. We achieve this by establishing a connection to the problem of document exchange under Hamming distance with asymmetric information, as follows.

Intuitively, in Bob's process of recovering the string x, in each level Bob keeps track of the positions of the possible blocks where his version $\tilde{x}$ and x may be different (we call these blocks bad). More specifically, recall that we can show in each level, with high probability there are at most $O(k)$ bad blocks. In the next level the number of these blocks will at most double due to splitting, however since we use random hash functions with $O(1)$ output bits, we can show that in the next level with high probability Bob will detect $O(k)$ bad blocks and update them. Some of the updated blocks may still be bad, but Bob knows the positions of all updated blocks, and he also knows that there are at most $O(k)$ bad blocks in them after the update. Now, suppose these updates happen in level j, and Bob is now in level $i > j$. Then the $O(k)$ updated blocks will split into $O(k2^{i-j})$ smaller blocks. If any of these smaller blocks is bad and it remains undetected so far, then it must have gone through $j - i$ different hash functions. If we choose all hash functions independently, then the probability that this happens is $2^{-c(i-j)}$ for some constant c. By choosing the number of output bits of the hash functions to be a large enough constant, we know that the expected number of smaller bad blocks that remain undetected so far is $O(k/2^{i-j})$. With a little extra effort, we can show that with high probability the number of these blocks is at most $k_{i-j} = \max\{k/\log^3 \frac{n}{k}, 20k/2^{i-j}\}$, and Bob knows that these blocks are inside the subset S_{i-j} with size $O(k2^{i-j})$, which stems from the $O(k)$ updated blocks in level j. In other words, this gives a forest with the $O(k)$ updated blocks in level j being the roots, and the at most k_{i-j} bad blocks are among the $|S_{i-j}| = O(k2^{i-j})$ leaves.

Note that the bad blocks in level i can come from the updated blocks in all previous levels, thus we get a vector $\mathbf{S} = (S_1, \cdots, S_{i-1})$ and a vector $\mathbf{k} = (k_1, \cdots, k_{i-1})$. Furthermore in this process, whenever a bad block stemming from some level j gets detected and updated in a later level j', this new block in level j' will become a new root and all its descendents are removed from the set S_{i-j} and put into the set $S_{i-j'}$. This ensures that the final subsets $(S_1, \cdots, S_{i-1})$ are disjoint. Finally, only Bob knows the sets $(S_1, \cdots, S_{i-1})$, but both parties know $(s_1 = |S_1|, \cdots, s_{i-1} = |S_{i-1}|)$ and $(k_1, \cdots, k_{i-1})$. Thus, we have reduced the problem of sending the sketch in level i to the problem of document exchange under Hamming distance with asymmetric information.

1.3.2 Document exchange for a special setting of parameters We now give our protocol for document exchange with asymmetric information, in the special setting described above. Recall that we have $s_i = O(k2^i), k_i = \max\{20k/2^{i-1}, k/\log^3 \frac{n}{k}\}, i \in [t], t = O(\log \frac{n}{k})$. One can compute $\mathsf{H}(\mathbf{s}, \mathbf{k}) = \Theta(k)$ here, so our protocol for the general setting will result in sub-optimal communication complexity. We give a different protocol here, which uses just one expander graph instead of a sequence of expander graphs.

The expander graph $G : [n] \times [d] \to [m]$ is generated by the shared randomness, with $m = O(k)$ and the following expansion property: for every $R \subseteq \cup_{i=1}^{t} S_i$ where $|R| \in [k/\log \frac{n}{k}, O(k)]$ and $\forall i \in [t], |R \cap S_i| \leq 20k_i$,

we have $|\Gamma(R)| \geq 0.9d|R|$. Limiting the expansion to restricted sets rather than all sets R with $|R| \in [k/\log\frac{n}{k}, O(k)]$ is the key to reduce the number of right vertices from $\Omega(k\log\frac{n}{k})$ to $O(k)$. Indeed, using a careful analysis of probabilities, we show that a random bipartite graph with constant d and $m = O(k)$ satisfies this property with high probability. The main intuition is that the sequence $\{s_i, i \in [t]\}$ roughly increases exponentially, while the sequence $\{k_i, i \in [t]\}$ roughly decreases exponentially.

Using this expander Alice sends her parity checks to Bob, and Bob again runs a belief propagation algorithm. The purpose of this phase is to reduce the total Hamming distance between x and $\tilde{x}$ (Bob's current version of x, starting with $\tilde{x} = y$) to at most $k/\log\frac{n}{k}$. However, the belief propagation has tricky issues here, as the standard approach may flip much more than $20k_i$ bits in S_i. This can result in a subset $R \subseteq [n]$ which does not have good expansion, thus ruining the whole process. To fix this, we prohibit the algorithm from flipping more than $20k_i$ bits in S_i for each i. This is done by keeping track of the number of already flipped bits in each S_i, and for any i if this number reaches $19k_i$, then subsequently in S_i the algorithm will only flip bits that are previously flipped.

To show that this indeed works, at each step of the belief propagation, let $R \subseteq \cup_{i=1}^{t} S_i$ stand for the set of indices where x and $\tilde{x}$ have different bits, and let R' stand for R restricted to the indices which we can flip (due to our modification). Thus R' always has good expansion. Our first observation is that at any time, $|R'| \geq 0.9|R|$. This is because R' is different from R only if for some S_i, the number of bits already flipped is at least $19k_i$. However originally there are at most k_i errors in S_i, so we have introduced at least $18k_i$ new errors. This means $\forall i, |R' \cap S_i| \geq 0.9|R \cap S_i|$, and thus $|R'| \geq 0.9|R|$. Now let $(\mathsf{s}', \mathsf{u}')$ and (s, u) be the number of satisfied and unsatisfied checks in $\Gamma(R')$ and $\Gamma(R)$ respectively. We know $\mathsf{s}' + \mathsf{u}' \geq 0.9d|R'|$. Also, again by the fact that each satisfied check in $\Gamma(R)$ has at least two neighbors in R, we have $2\mathsf{s}' + \mathsf{u}' \leq 2\mathsf{s} + \mathsf{u} \leq d|R| \leq \frac{10}{9}d|R'|$. From these two inequalities we can still deduce that $\mathsf{u}' \geq 0.7d|R'|$, thus Bob can find a bit in R' to flip.

When this process stops, the Hamming distance between x and $\tilde{x}$ is at most $k/\log\frac{n}{k}$. We can now use a deterministic document exchange protocol for Bob to recover x. The communication complexity is $O((k/\log\frac{n}{k})\log\frac{n}{k}) = O(k)$. The only error probability here comes from the generation of the expander graph, which is $2^{-\Omega(k/\log\frac{n}{k})}$. We also show that the other errors in the protocol for edit distance is $2^{-\Theta(k/\log^3\frac{n}{k})}$. Thus the total error of the protocol for edit distance is $2^{-\Theta(k/\log^3\frac{n}{k})}$. When $k < \log^4 n$, we can switch to the protocol in [20] which has error $1/\mathsf{poly}(n)$.

2 Discussion and Open Problems

In this paper we initiated a systematic study of document exchange and error correcting codes with asymmetric information. While we provided both lower bounds and upper bounds, as well as efficient randomized constructions that are close to optimal, there are still many interesting problems left. We list some below.

Question 1: The most obvious open problem is to achieve optimal communication complexity (i.e., $\mathsf{H}(\mathbf{s}, \mathbf{k})$) for a one round randomized protocol. Two related questions are to reduce the error probability of the randomized protocol, and to study the case where the condition $\forall i, s_i \geq 2k_i$ does not hold. For example, is there a better deterministic protocol for the latter case?

Question 2: A better understanding of the problem in the case of two sided asymmetric information. In the full version of this paper we study the case of two sided asymmetric information where $s^A + s^B \leq n$, i.e., the subsets from both parties can be disjoint in the worst case. For this case we can get an upper bound close to the lower bound. What happens when $s^A + s^B > n$? In this case the subsets from both parties are guaranteed to overlap, and the situation becomes more complicated.

Question 3: Two round deterministic protocol. We showed that for any one round deterministic protocol, the asymmetric information is not useful. However, by a result of Orlitsky [31], there exists a two round exponential time deterministic protocol with communication complexity $O(\mathsf{H}(\mathbf{s}, \mathbf{k}) + \log n)$. It is an interesting open problem to see if we can design efficient protocols matching this bound.

Question 4: Optimal deterministic document exchange under edit distance. Our results also bring some hope to obtain an optimal deterministic document exchange protocol under edit distance. Especially, we have replaced the decoding by exhaustive search approach in [20] by an efficient decoding algorithm. However, how to appropriate pick a hash function remains a problem.

Paper Organization

The rest of the paper is organized as follows. In Section 3 we introduce some basic technical tools. In Section 4 we show lower bounds for asymmetric DE in the general

setting. In Section 5 we give our protocol for asymmetric DE in the general setting. In Section 6 we give our protocol for asymmetric DE in a special setting.

3 Preliminaries

3.1 Error correcting codes We will use the following well known parity check computation based on bipartite expander graphs.

CONSTRUCTION 3.1. ([35]) *Let $\Gamma : [n] \times [d] \to [m]$ be a bipartite graph with n left vertices, m right vertices, left degree d. The encoding of the Γ-expander code, on input message $x \in \{0,1\}^n$, is computed as*

$$x \circ z,$$

where $z \in \{0,1\}^m$, $z[i] = \bigoplus_{j \in \Gamma^{-1}(i)} x[j], i \in [m]$.

DEFINITION 3.2. ([19]) *A bipartite graph with n left vertices, m right vertices and left degree d is a (k, a) expander if for every set of left vertices $S \subseteq [n]$ of size k, we have $|\Gamma(S)| > ak$. It is a $(\leq k_{\max}, a)$ expander if it is a (k, a) expander for all $k \leq k_{\max}$.*

Here $\forall x \in [n]$, $\Gamma(x)$ outputs the set of all neighbours of x. It is also a set function which is defined accordingly. Also $\forall x \in [n], y \in [d]$, the function $\Gamma : [n] \times [d] \to [m]$ is such that $\Gamma(x, y)$ is the y-th neighbour of x.

THEOREM 3.1. ([19]) *For all constants $\alpha > 0$, for every $n \in \mathbb{N}$, $k_{\max} \leq n$, and $\epsilon > 0$, there exists an explicit $(\leq k_{\max}, (1-\epsilon)d)$ expander with n left vertices, m right vertices, left degree $d = O((\log n)(\log k_{\max})/\epsilon)^{1+1/\alpha}$ and $m \leq d^2 k_{\max}^{1+\alpha}$. Here d is a power of 2.*

The explicitness here means, given a left node, and an edge, the induced right node computed found in time $O(\log n + \log d)$.

THEOREM 3.2. ([35]) *Let $\Gamma : [n] \times [d] \to [m]$ be a $(\leq k, 3/4d)$ bipartite graph with left degree d_l, right degree d_r. Let y be an n-bit string whose distance from a codeword x is at most $k/2$. Then a repeated application of the following decoding algorithm to y will return x in time $O(d_l d_r m)$.*

Decoding algorithm: Upon receiving the input n-bit string y, as long as there exists a variable such that most of its neighbouring constraints are not satisfied, flip it.

THEOREM 3.3. ([24] [16] [34]) *There exists an explicit construction of algebraic geometry linear $(n, m, d)_q$-code with $d + m \geq n - \frac{n}{\sqrt{q}-1.1}, q = \lceil \frac{n}{d} \rceil^2$, polynomial-time decoding when the number of errors is less than half of the distance. Here n, q should be at least some fixed constants.*

Moreover for every message $x \in \mathbb{F}_q^m$, the codeword is $x \circ z$ for some redundancy $z \in \mathbb{F}_q^{n-m}$. In other words, the code is systematic.

3.2 Pseudorandomness A distribution X over Σ^n is k-wise independent if for any k variables in X, their marginal distribution is uniform.

THEOREM 3.4. *There exists an explicit construction of κ-wise independence generator $g : \{0,1\}^s \to \{0,1\}^n$, where $s = O(\kappa \log \frac{n}{\kappa})$.*

Proof. Let $C^\perp$ be an algebraic geometry linear $(n, m, d)_q$-code constructed by Theorem 3.3, with $d = \kappa + 1$, $m \geq n - O(\kappa)$, $q = \mathsf{poly}(n/d) = \mathsf{poly}(n/\kappa)$.

Consider the dual code $C = (C^\perp)^\perp$. By duality of codes, its message length is $n - m = O(\kappa)$. Let the generator be $g(\cdot) = C(\cdot)$, i.e. the encoding function of C. Note that the seed length in bits is $s = (n-m)\log q = O(\kappa \log \frac{n}{\kappa})$.

We claim that any κ columns of the generating matrix $M \in \mathbb{F}_q^{m \times n}$ of C, are linearly independent. Since otherwise there will be a codeword in $C^\perp$, which has hamming distance $\leq \kappa = d - 1$ from the codeword 0-vector.

Next we show $g(u) = uM$ is κ-wise independent, when u is uniform. For any κ symbols in the output, the corresponding κ columns of M are linearly independent. So the matrix $M_K, K = \{\text{indices of these } \kappa \text{ columns}\}$, formed by these columns has rank κ. Thus there are κ rows which are linearly independent. Hence each linear combination of these κ rows in M_K can uniquely represent one vector in the space of κ symbols. So $(uM)_K$ is uniform.

To see this is an explicit construction, note that the encoding of $C^\perp$ is explicit. So the encoding of each $e_i \in \mathbb{F}_q^m, i \in [m]$, where e_i is i-th unit vector, is explicit. Thus the encoding matrix $M^\perp$, whose i-th row is $C^\perp(e_i)$, can be computed explicitly. The corresponding parity check matrix, which is actually M the encoding matrix of its dual code C, can be computed explicitly using $M^\perp$ by standard procedures. ∎

Random variables $X_1, X_2, \ldots, X_n \in \{0,1\}^n$ are ε-almost κ-wise independent in max norm if

$$\forall i_1, i_2, \ldots, i_\kappa \in [n], \forall x \in \{0,1\}^\kappa,$$
$$\left|\Pr[X_{i_1} \circ X_{i_2} \circ \cdots \circ X_{i_\kappa} = x] - 2^{-\kappa}\right| \leq \varepsilon.$$

A function $g : \{0,1\}^d \to \{0,1\}^n$ is an ε-almost κ-wise independence generator in max norm if $g(U) =$

$X = X_1 \circ \cdots X_n$ are ε-almost κ-wise independent in max norm. Unless stated otherwise, we only consider max norm in the following context.

THEOREM 3.5. ([4]) *There exists an explicit construction s.t. for every $n, \kappa \in \mathbb{N}$, $\varepsilon > 0$, it computes an ε-almost κ-wise independence generator $g : \{0,1\}^d \to \{0,1\}^n$, where $d = O(\log \frac{\kappa \log n}{\varepsilon})$.*

This construction is highly explicit in the sense that, $\forall i \in [n]$, the i-th output bit can be computed in time $\tilde{O}(\log n + \log \frac{1}{\varepsilon})$ given the seed and i. (The $\tilde{O}$ here hides some $\log\log n$, $\log\log(1/\varepsilon)$ factors)

Another tool will use is the general moment inequality for k-wise independence.

THEOREM 3.6. *Let $X_i \in \{0,1\}, i = 1, \ldots, n$, be a sequence of k-wise independent random variables. Let $X = \sum_{i=1}^n X_i$.*

For every $\varepsilon > 0$,

$$\Pr[X \geq (1+\varepsilon)\mathbb{E}X] \leq \left(\frac{1}{1+\varepsilon}\right)^k.$$

3.3 LCS and Matching Consider two strings $x \in \{0,1\}^{pn}, y \in \{0,1\}^{n'}$, hash functions $h_j : \{0,1\}^p \to \{0,1\}^q, j \in [n]$. A monotone matching $w = ((\rho_1, \rho'_1), \ldots, (\rho_{|w|}, \rho'_{|w|}))$ between x, y under $h_j, j \in [n]$ is s.t. for every $i \in [|w|]$, $h_{\rho_i}(x[\rho_i, \rho_i + p)) = h_{\rho_i}(y[\rho'_i, \rho'_i + p))$, where $\rho_i \in [pn], \rho'_i \in [n']$. Also we consider x as being cut into length p blocks and each ρ_i has to be a starting position of a block in x.

LEMMA 3.3. *For any $x \in \{0,1\}^{pn}, y \in \{0,1\}^{n'}, k \in \mathbb{N}$, $S \subseteq [n], |S| = s$, $h_j : \{0,1\}^p \to \{0,1\}^q, j \in [n]$, the number of matchings $w = ((\rho_1, \rho'_1), \ldots, (\rho_{|w|}, \rho'_{|w|}))$ between x_S and y under $h_j, j \in [n]$ s.t. $|\rho'_1 - \rho_1| + |(\rho'_2 - \rho'_1) - (\rho_2 - \rho_1)| + \cdots + |(\rho'_{|w|} - \rho'_{|w|-1}) - (\rho_{|w|} - \rho_{|w|-1})| \leq k$, is at most $2^{2s + k(\log \frac{k+s-1}{k} + \log e)}$.*

Here x_S refers to the sequence of blocks of x. The j-th block of it is $x_S[j] \in \{0,1\}^p, j \in [s]$. We use $\mathsf{pos}(j)$ to refer to the starting position of block $x_S[j]$ in x. The proof is deferred to the full version.

LEMMA 3.4. (DP FOR LCS WITHIN k EDIT OPERATIONS) *There is an algorithm, on input $x \in \{0,1\}^{pn}, y \in \{0,1\}^{n'=O(np)}, S \subseteq [n], k = \mathsf{ED}(x, y)$, hash functions $h_i : \{0,1\}^p \to \{0,1\}^q, i \in [n]$, outputs a monotone matching $w = ((u_1, u'_1), \ldots, (u_{|w|}, u'_{|w|}))$ between x_S and y under $h_i, i \in [n]$ s.t. $|w| \geq |S| - k$, and $|u'_1 - u_1| + |(u'_2 - u'_1) - (u_2 - u_1)| + \cdots + |(u'_{|w|} - u'_{|w|-1}) - (u_{|w|} - u_{|w|-1})| \leq k$.*

The proof is deferred to the full version.

4 Negative Result

In this section, we show some lower bounds for the asymmetric document exchange and error correcting codes. Given the vectors $\mathbf{s} = (s_1, \cdots, s_t)$ and $\mathbf{k} = (k_1, \cdots, k_t)$, we define

$$\mathsf{H}(\mathbf{s}, \mathbf{k}) = \log\left(\prod_{i=1}^t \left(\sum_{j=0}^{k_i} \binom{s_i}{j}\right)\right) = \sum_{i=1}^t \log\left(\sum_{j=0}^{k_i} \binom{s_i}{j}\right).$$

Similarly, for two integers s and k with $s \geq k$, we define

$$\mathsf{H}(s, k) = \log\left(\sum_{j=0}^{k} \binom{s}{j}\right).$$

Note that in particular we have $\mathsf{H}(\mathbf{s}, \mathbf{k}) \geq \sum_{i=1}^t k_i \log(s_i/k_i)$ and $\mathsf{H}(s, k) \geq k \log(s/k)$.

We now have the following theorems.

THEOREM 4.1. *In an $(\mathbf{s}, \mathbf{k}, t)$ asymmetric DE problem where Bob has the vector of subsets $\mathcal{S} = (S_1, \cdots, S_t)$, let $k = \sum_{i=1}^t k_i$ and suppose Alice learns Bob's string. Then any deterministic protocol has communication complexity at least $\mathsf{H}(n, k)$, and any randomized protocol with success probability $\geq 1/2$ has communication complexity at least $\mathsf{H}(n, k) - 1$. This holds even if Alice knows $\mathbf{s}$ and $\mathbf{k}$.*

The proof is deferred to the full version.

We now consider the case where Bob tries to learn Alice's string, and we have the following theorem.

THEOREM 4.2. *In an $(\mathbf{s}, \mathbf{k}, t)$ asymmetric DE problem where Bob has the vector of subsets $\mathcal{S} = (S_1, \cdots, S_t)$, let $k = \sum_{i=1}^t k_i$ and suppose Bob learns Alice's string. Then any randomized protocol with success probability $\geq 1/2$ has communication complexity at least $\mathsf{H}(\mathbf{s}, \mathbf{k}) - 1$. Furthermore if $\forall i, s_i = |S_i| \geq 2k_i$, then any one round deterministic protocol has communication complexity at least $\mathsf{H}(n, k)$. This holds even if Alice knows $\mathbf{s}$ and $\mathbf{k}$.*

The proof is deferred to the full version.

We also have the following theorem for asymmetric error correcting codes.

THEOREM 4.3. *In an $(\mathbf{s}, \mathbf{k}, t)$ asymmetric ECC problem where Bob has the vector of subsets $\mathcal{S} = (S_1, \cdots, S_t)$, let $k = \sum_{i=1}^t k_i$. If $\forall i, s_i = |S_i| \geq 2k_i$, then any deterministic code must have distance at least $2k + 1$. In particular, $m \leq n - \mathsf{H}(n, k)$. Furthermore, any randomized code with success probability $\geq 1/2$ must have message length $m \leq n - \mathsf{H}(\mathbf{s}, \mathbf{k}) + 1$.*

The proof is deferred to the full version.

5 Document Exchange and Error Correcting Codes with Asymmetric Information in the General Setting

We give a random protocol for the general setting s.t. the communication complexity is close to optimal.

5.1 Key components

LEMMA 5.1. *For every $S \subseteq [n]$, integer $k_0 \le k \le s = |S|$, the probability that a random bipartite graph with n left vertices, $m \ge 2dk2^{1/\delta}$ right vertices, left degree $d = O(\log \frac{2s}{k})$, having*

$$\begin{aligned} &\text{for every } R \subseteq S, \text{ with } |R| \in [k_0, k] \\ &|\Gamma(R)| > (1-\delta)d|R|, \end{aligned} \tag{5.2}$$

is at least $1-\varepsilon$, where $\varepsilon = 2^{-\Theta(\delta(\log \frac{2s}{k})k_0 \log \frac{2k}{k_0})}$.

Note that when $k_0 = 1$, we get an $(n, m, d, S, \le k, 1-\delta)$ expander with probability at least $1 - 2^{-\Theta(\delta \log \frac{2s}{k} \log(2k))} \le 1 - 1/\mathsf{poly}(s)$.

We also denote a bipartite graph with the expansion property stated as an $(n, m, d, S, [k_0, k], 1-\delta)$ expander.

Proof. The total number of sets R with size r is at most $(\frac{es}{r})^r$.

For a fixed set R, a fixed set $T \subseteq [m], |T| = (1-\delta)d|R|$

$$\Pr[\Gamma(R) \subseteq T] = \left(\frac{|T|}{m}\right)^{dr} = \left(\frac{(1-\delta)dr}{m}\right)^{dr}. \tag{5.3}$$

There are at most

$$\binom{m}{|T|} \le \left(\frac{em}{|T|}\right)^{|T|} = \left(\frac{em}{(1-\delta)dr}\right)^{(1-\delta)dr} \tag{5.4}$$

such set T.

So by a union bound, the probability that for every $R, |R| = r$, $\Gamma(R) \le (1-\delta)dr$ is at most

$$\begin{aligned} &\left(\frac{em}{(1-\delta)dr}\right)^{(1-\delta)dr} \times \left(\frac{(1-\delta)dr}{m}\right)^{dr} \times (\frac{es}{r})^r \\ =&e^{(1-\delta)dr}\left(\frac{(1-\delta)dr}{m}\right)^{\delta dr} \times (\frac{es}{r})^r \\ \le&e^{dr}e^{-\delta dr \log \frac{m}{dr}}(\frac{es}{r})^r \\ \le&2^{-\Theta(\delta dr \log \frac{2k}{r})}. \end{aligned} \tag{5.5}$$

by letting $m = 2dk2^{1/\delta}$, $d = O(\log \frac{2s}{k})$.

By another union bound the probability that for every $R, |R| \in [k_0, k]$, it does not have a good expansion is at most $\sum_{j=k_0}^{k} 2^{-\Theta(\delta dj \log \frac{2k}{j})} \le (k - k_0 + 1)2^{-\Theta(\delta dk_0 \log \frac{2k}{k_0})} \le 2^{-\Theta(\delta dk_0 \log \frac{2k}{k_0})}$.

When $k_0 = 1$, this is at most $2^{-\Theta(\delta \log \frac{2s}{k} \log(2k))} \le 2^{-\Theta(\delta \log \frac{2s}{k} \log(2k))} \le 1/\mathsf{poly}(s)$. ∎

LEMMA 5.2. *Assume Γ is an $(n, m, d, S, [k_1', 2k_1], 0.9)$ expander. Let y be the expander-code encoding of x using Γ. Then there is an explicit decoding which, on input x' which has $k_i, i \in [t]$ errors in S_i from x, with $k_1 \ge k_1' \ge c\sum_{i=2}^{t} k_i$, $c = 10$, outputs $\tilde{x}$ that has at most k_1' errors in S_1.*

Proof. We propose the following algorithm. For every iteration, find the first bit in S_1 s.t. it has more unsatisfied checks than satisfied ones. Loop until we cannot find such bit anymore.

Now we show this works. Assume there are at least k_1' errors in S_1. Denote A as the set of indices of these errors. Let s be the number of satisfied neighbors of $A_1 = A \cap S_1$. Let u be the number of unsatisfied neighbors of A_1. By the expander property, $|\Gamma(A)| \ge 0.9d|A_1|$. So

$$s + u \ge 0.9d|A_1|. \tag{5.6}$$

On the other hand, each satisfied check is connected to at least one vertex in A_1 since it is in $\Gamma(A_1)$. Thus it has to be connected to at least 2 vertices in A to make it to be satisfied. Also each unsatisfied check is connected to at least 1 vertex in A_1. Hence

$$2s + u \le d|A| \le d|A_1| + d\sum_{i=2}^{t} k_i \le (1 + \frac{1}{c})d|A_1|. \tag{5.7}$$

By Equation (5.6) and Equation (5.7),

$$u \ge 0.8d|A_1|.$$

So there has to be ≥ 0.1 fraction of vertices in S_1 having more unsatisfied checks than satisfied ones. As a result, the algorithm can find a bit to flip and u is decreasing. On the other hand, if at some iteration, $|A_1| = 2k_1$, then $u \ge 1.6dk_1$ but initially $u \le dk_1$ which contradicts that u is decreasing. As a result, the iterations will continue until there are less than k_1' errors in S_1. ∎

THEOREM 5.1. *There is an efficient 1-round protocol s.t. for every (s, k) DE problem, it has communication complexity $O(k \log \frac{2s}{k})$, success probability $1 - 2^{-\Theta(\log \frac{2s}{k} \log k)}$.*

It is implied by Lemma 5.1 Lemma 5.2.

5.2 The protocol

Without loss of generality, we assume $k_1 \ge k_2 \ge \cdots \ge k_t$.

THEOREM 5.2. *There is a 1-way efficient protocol s.t. for every $(\mathbf{s}, \mathbf{k}, t)$ DE with $k_i \le s_i/2, \forall i \in [t]$, it has success probability $1 - 2^{-\Omega(k_t)} - 1/\mathsf{poly}(s)$, communication complexity $O(t^2 \sum_{i \in [t]} k_i \log \frac{s_i}{k_i})$.*

CONSTRUCTION 5.3. *Efficient protocol for* $(\mathbf{s}, \mathbf{k}, t)$ *DE* .

Alice: on input x,

1. *Let* $i' = 0, k' = 0$, *string* z *be empty string;*
   - 1.1. *While* $i' \leq t-2$, *find* $i > i'$ *s.t.* $k' + \sum_{j=i'+1}^{i} k_j > k''$, *where* $k'' = c\sum_{j=i+1}^{t} k_j$; *If cannot find* i *then break the iterations;*
   - 1.2. *Generate an* $(n, m, d, \cup_{j=1}^{i} S_j, [k'', 2(k' + \sum_{j=i'+1}^{i} k_j)], 0.9)$*-expander* Γ *by Lemma 5.1, where* $d = O(\log \frac{\sum_{j=1}^{i} s_j}{k' + \sum_{j=i'+1}^{i} k_j})$;
   - 1.3. *Compute* z_i *which is the expander code of* x *using* Γ, $z = z \circ z_i$;
   - 1.4. *Let* $i' = i, k' = k''$.
2. *Encode* x *to be* z_{final} *by using a* $(n, m, d = O(\log \frac{s}{k_{\text{final}}}), S, \leq 2k_{\text{final}}, 0.9)$ *expander* Γ_{final} *generated by Lemma 5.1, where* $k_{\text{final}} = k' + \sum_{j=i'+1}^{t} k_j$;
3. *Send* $z \circ z_{\text{final}}$ *to Bob.*

Bob: on input y, $\mathbf{S}, \mathbf{k}$, *together with the message* $z \circ z_{\text{final}}$ *from Alice;*

1. *Let* $i' = 0, k' = 0, y' = y$;
   - 1.1. *While* $i' \neq t$, *find* $i > i'$ *s.t.* $k' + \sum_{j=i'+1}^{i} k_j > k''$, *where* $k'' = c\sum_{j=i+1}^{t} k_j, c = 10$;
   - 1.2. *Generate an* $(n, m, d, \cup_{j=1}^{i} S_j, [k'', 2(k' + \sum_{j=i'+1}^{i} k_j)], 0.9)$*-expander* Γ *by Lemma 5.1 using the same randomness as of Alice;*
   - 1.3. *Use* Γ, z_i *to reduce the number of errors of* y *in* $\cup_{j=1}^{i} S_j$ *to be at most* k'' *by Lemma 5.2;*
   - 1.4. *Let* $i' = i, k' = k''$.
2. *Decode* x *by Lemma 5.2 for the* $(S, k' + k_t)$ *setting, using* y', z_{final}, *and the expander generated the same as the* Γ_{final} *of Alice;*

LEMMA 5.4. *The communication complexity is* $O\left(t^2 \sum_{j\in[t]} k_j \log \frac{s_j}{k_j}\right)$.

The proof is deferred to the full version.

Next we show the correctness.

LEMMA 5.5. *Bob can compute* x *correctly with probability at least* $1 - 2^{-\Omega(k_t)} - 1/\mathsf{poly}(s)$.

Proof. In the first iteration, since Γ is an $(n, m, d, \cup_{j=1}^{i} S_j, [c\sum_{j=i+1}^{t} k_j, 2(\sum_{j=1}^{i} k_j)])$ expander, by Lemma 5.2, we can successfully reduce the number of errors in $\cup_{j=1}^{i} S_j$ to be $\leq k''$.

Note that as long as $k_{i'+1} > 0$, the number i, found in the iteration, will be $i' + 1$. So the iteration will continue until $i' = t-1$. After the iterations, the number of errors in S is at most $k' + k_t = (c+1)k_t$.

Finally, using z_{final} and Γ_{final}, by Lemma 5.2, Bob can compute x correctly.

The protocol succeeds once all random expander graphs are as desired. For random expander graph in iteration i, the success probability is $1 - 2^{-\Omega(dk'' \log \frac{2k'}{k''})} \leq 1 - 2^{-\Omega(dk'')}$, by Lemma 5.1. So by a union bound, the probability, that all iterations success, is at least $1 - 2^{-\Omega(k_t)}$. In the final step, the success probability is $1 - \frac{1}{\mathsf{poly}(s)}$ by Theorem 5.1. Hence the final success probability is as desired. ∎

Proof. [Proof of Theorem 5.2]

The correctness and communication complexity immediately follows from Lemma 5.4, Lemma 5.5.

For the efficiency, note that in Alice's algorithm, she just randomly generate a bipartite graph with logarithmic degree. And apply the expander encoding to get the sketch. So this is in near linear time. For Bob's algorithm, as $S_i, i \in [t]$ are disjoint, and the belief propagation can be done in near linear time. Other operations are also in near linear time. So Bob's algorithm is also in near linear time. ∎

When t is large, we can group some sets together to reduce t and hence get the following theorem.

THEOREM 5.3. *There is a* 1*-way efficient protocol s.t. for every* $(\mathbf{s}, \mathbf{k}, t)$ *DE with* $k_i \leq s_i/2, \forall i \in [t]$, *it has success probability* $1 - 2^{-\Omega(k_t)} - 1/\mathsf{poly}(s)$, *communication complexity* $O\left(\chi^2(\mathbf{s}, \mathbf{k}, t) \sum_{i\in[t]} k_i \log \frac{s_i}{k_i}\right)$.

The running time of both parties are $\tilde{O}(n)$.

The proof is deferred to the full version.

Notice that χ can only be as large as $O(\log \log n)$. So we have the following corollary.

COROLLARY 5.1. *There is a* 1*-way efficient protocol s.t. for every* $(\mathbf{s}, \mathbf{k}, t)$ *DE with* $k_i \leq s_i/2, \forall i \in [t]$, *it has success probability* $1 - 2^{-\Omega(k_t)} - 1/\mathsf{poly}(s)$, *communication complexity* $O\left(\log^2 \log n \sum_{i\in[t]} k_i \log \frac{s_i}{k_i}\right)$.

The running time of both parties are $\tilde{O}(n)$.

5.3 From DE to stochastic coding We show that our construction for DE can be modified to work for stochastic coding setting.

THEOREM 5.4. *There is an efficient stochastic ECC s.t. for every $(\mathbf{s},\mathbf{k},t)$ type errors with $k_i \le s_i/2, \forall i \in [t]$, it has success probability $1-2^{-\Omega(k_t)}-1/\mathsf{poly}(s)$, message length $n - O\left(\chi^2(\mathbf{s},\mathbf{k},t)\mathsf{H}(\mathbf{s},\mathbf{k})\right)$.*

The running time of both encoding and decoding are $\tilde{O}(n)$.

The proof is deferred to the full version.

6 Document Exchange with Asymmetric Information in a Special Setting

We first develop a randomized two-party (Alice and Bob) one-way hamming error document exchange protocol in which Bob knows the errors can only happen in some subsets of all positions, where in each subset the number of errors is also bounded.

The reason we consider this kind of encoding/decoding for special error patterns is that it can have shorter redundancy than the general coding for bounded number of hamming errors.

The encoding utilize a randomized bipartite expander graph with a large expansion.

LEMMA 6.1. *For every $n,k,k',k'',r,d,t \in \mathbb{N}$, $k' \le r \le k \le n$, $k''t\log\frac{ek2^t}{k''} \le k'\log\frac{2k}{k'}$, $\delta \in (0,1)$, $d \ge \delta^{-1}$, constant $c>0$, disjoint sets $S_i \subseteq [n], i \in [t], |S_i| = k2^{O(i)}$, $\overline{k}_i = \max(k/2^{O(i)}, k'') \le |S_i|/2$, the probability that a random bipartite graph with n left vertices, $m \ge 2dk2^{1/\delta}$ right vertices, left degree d, having that*

for every $R \subseteq \cup_{i\in[t]}S_i, |R| = r \ge k'$, with $|R\cap S_i| \le \overline{k}_i$, every $i \in [t]$,
it holds $|\Gamma(R)| > (1-\delta)dr$,

is at least $1-\varepsilon$, where $\varepsilon = 2^{-\Theta(\delta dk'\log\frac{2k}{k'})}$.

We denote the generated expander graph as a $(n,m,d,\mathbf{S},\overline{\mathbf{k}},[k',k],1-\delta)$-expander, where $\overline{\mathbf{k}}$ is the sequence of all $\overline{k}_i, i\in[t]$.

Proof. We show that a uniformly sampled bipartite graph works. The bipartite graph with n left vertices, m right vertices, left degree d, is generated as follows. Each edge, from one vertex of the left, has its ending vertex being uniformly chosen from the right vertices.

For a fixed R, if $|\Gamma(R)| \le (1-\delta)dr$, then there exists a set $T \subseteq [m]$ s.t. $|T| = (1-\delta)dr, |\Gamma(R)| \subseteq T$. There are at most

$$\binom{m}{|T|} \le \left(\frac{em}{|T|}\right)^{|T|} = \left(\frac{em}{(1-\delta)dr}\right)^{(1-\delta)dr} \tag{6.8}$$

such set T. For each T,

$$\Pr[\Gamma(R)\subseteq T] = \left(\frac{|T|}{m}\right)^{dr} = \left(\frac{(1-\delta)dr}{m}\right)^{dr}. \tag{6.9}$$

Consider a fixed r. Assuming $r \in [k_{j+1},k_j]$, for some $j\in[t]$. Notice that $j = \Theta(\log\frac{2k}{r})$.

Let $r_i = R\cap S_i$. The total number of different sequences $r_1,\ldots,r_t$ is at most

$$\binom{r+t}{r} \le \left(\frac{e(r+t)}{r}\right)^r \le \left(O(\frac{2k}{r})\right)^r \le 2^{O(r\log\frac{2k}{r})}. \tag{6.10}$$

Consider a fixed sequence $r_i, i\in[t]$ with $r_i \le \overline{k}_i$. The total number of possibilities of $R\cap S_j,\ldots,R\cap S_t$ is at most $\prod_{i=j}^t\binom{|S_i|}{r_i} \le \prod_{i=j}^t\binom{|S_i|}{\overline{k}_i} \le \prod_{i=j}^t\left(\frac{e|S_i|}{\overline{k}_i}\right)^{\overline{k}_i} \le \prod_{i=j}^{t'}2^{O(i\frac{k}{2^{O(i)}})}\cdot\prod_{i=t'}^t\left(\frac{e|S_t|}{k''}\right)^{k''} = 2^{O\left(\sum_{i=j}^{t'}i\frac{k}{2^{O(i)}}\right)}\cdot 2^{O((t-t')k''\log\frac{ek2^t}{k''})} \le 2^{O(\frac{k}{2^{O(j)}}j)}\cdot 2^{O(r\log\frac{2k}{r})} = 2^{O(r\log\frac{2k}{r})}$. Here t' is the first index s.t. $\overline{k}_i = k''$.

On the other hand, the total number of possibilities of $R\cap S_1,\ldots,R\cap S_j$ is at most $\prod_{i=1}^j\binom{|S_i|}{r_i} \le \binom{\sum_{i=1}^j|S_i|}{\sum_{i=1}^j r_i} \le \binom{O(k2^{O(j)})}{\sum_{i=1}^j r_i} \le \binom{O(k2^{O(j)})}{r} \le \left(O(\frac{k2^{O(j)}}{r})\right)^r \le 2^{O(r\log\frac{2k}{r})}$.

So by a union bound, the probability that for every $R, |R|=r, |R\cap S_i| \le \overline{k}_i, \Gamma(R) \le (1-\delta)dr$ is at most

$$\begin{aligned}
&\left(\frac{em}{(1-\delta)dr}\right)^{(1-\delta)dr}\times\left(\frac{(1-\delta)dr}{m}\right)^{dr}\times 2^{O(r\log\frac{2k}{r})}\\
=&e^{(1-\delta)dr}\left(\frac{(1-\delta)dr}{m}\right)^{\delta dr}\times 2^{O(r\log\frac{2k}{r})}\\
\le&e^{dr}e^{-\delta dr\log\frac{m}{dr}}2^{O(r\log\frac{2k}{r})}\\
\le&2^{-\Theta(\delta dr\log\frac{2k}{r})}
\end{aligned} \tag{6.11}$$

by letting $m = 2dk2^{1/\delta}$.

Since $k \ge r \ge k'$, it holds that $2^{-\Theta(\delta dr\log\frac{2k}{r})} \le 2^{-\Theta(\delta dk'\log\frac{2k}{k'})}$. □

The decoding algorithm has two parts. Both parts use belief propagation techniques. In the first part, we reduce the number of errors slightly by using z_1. In the second part, we further reduce the number of errors to 0 by using z_2.

CONSTRUCTION 6.2. *Let $n, m, d, t \in \mathbb{N}$, $k_i \in \mathbb{N}, k_i \leq n, i \in [t]$, $k' = O(k/\log \frac{n}{k})$, disjoint sets $S_i \subseteq [n], i \in [t]$. Let $S = \cup_{i\in[t]} S_i$.*

Let expander graph $\Gamma_1 : [n] \times [d_1] \to [m_1]$, s.t.

$$\forall R \subseteq \cup_{i\in[t]} S_i \text{ with } |R| \in [k', O(k)] \text{ and } \forall i \in [t], |R \cap S_i| \leq 20k_i, \text{ it holds } \Gamma_1(R) > 0.9d|R|.$$

Let C_0 be a systematic Algebraic Geometry code from Theorem 3.3, with alphabet $\mathbb{F}_q$, message length n/q, redundancy length $O(k')$ correcting $2k'$ errors.

Let $x \in \{0,1\}^n$ be the original message.

The decoding takes an input string $y \in \{0,1\}^n$, parity checks z_1 generated by expander encoding of x using Γ_1, and z_2 which is the redundancy part of $C_0(x)$.

Stage 1:

1. *(Generating the restriction set) Let $V = \emptyset$. For every $i \in [t]$, if the number of flipped bits in S_i is less than $19k_i$, then $V = V \cup S_i$ otherwise $V = V \cup \{j \mid$ the j-th bit is flipped previously by this algorithm$\}$; (If a bit is flipped twice, then it is regarded as not flipped)*
2. *Find $j \in V$ s.t. the number of unsatisfied parity checks in $\Gamma_1(j)$ is larger than $|\Gamma_1(j)|/2 = d_1/2$; Flip the j-th bit, and restart this stage; If no such j, go to the next step;*
3. *Go to the next stage.*

Stage 2 (classic belief propagation using z_2):

1. *Apply the decoding of C_0 on the current y concatenated with z_2.*
2. *Output the decoded message.*

LEMMA 6.3. *If $\mathsf{HD}(y_{\overline{S}}, x_{\overline{S}}) = 0, \forall i \in [l], \mathsf{HD}(y_{S_i}, x_{S_i}) \leq k_i$, then the decoder outputs x correctly.*

Proof. CLAIM 6.4. *The first stage ends in at most $O(m_1)$ rounds, and the number of errors in y is reduced to be less than $2k'$.*

Proof. Let A_τ be the set of indices of tampered bits (comparing to x) in y at (immediately before) the τ-th round. At the beginning $|A_1| = \mathsf{HD}(y, x)$.

We first show that if $|A_\tau| \geq 2k'$, then we can indeed find an index $j \in V$ s.t. the number of unsatisfied parity checks in $\Gamma_1(j)$ is larger than $|\Gamma_1(j)|/2$.

Denote $A'_\tau = A_\tau \cap V$. Let s, s' be the numbers of satisfied checks in $\Gamma_1(A_\tau), \Gamma_1(A'_\tau)$. Let u, u' be the numbers of unsatisfied checks in $\Gamma_1(A_\tau), \Gamma_1(A'_\tau)$.

Consider $i \in [l]$ s.t. the number of flipped bits is exactly $19k_i$. As $\mathsf{HD}(y_{S_i}, x_{S_i}) \leq k_i$, the number of tampered bits in S_i is at most $20k_i$. So $|A_\tau \cap S_i| \leq 20k_i$, since $\mathsf{HD}(y_{S_i}, x_{S_i}) \leq k_i$. Also note that these tempered bits (at the beginning of the stage) can be flipped by the algorithm, we know the current number of tampered bits in $A'_\tau \cap S_i$ is at least $18k_i$. So

$$|A'_\tau \cap S_i| \geq 0.9|A_\tau \cap S_i|. \tag{6.12}$$

For $i \in [l]$ s.t. the number of flipped bits is less than $19k_i$, since $V \cap S_i = S_i$,

$$|A'_\tau \cap S_i| = |A_\tau \cap S_i| \tag{6.13}$$

As a result, noting that $S_i, i \in [l]$ are disjoint,

$$\frac{|A'_\tau|}{|A_\tau|} = \frac{\sum_i |A'_\tau \cap S_i|}{\sum_i |A_\tau \cap S_i|} \geq 0.9, \tag{6.14}$$

As $|A_\tau| \geq 2k'$, it holds $|A'_\tau| \geq 1.8k' \geq k'$. By the expansion property of Γ_1,

$$s' + u' = |\Gamma_1(A'_\tau)| \geq 0.9d|A'_\tau|. \tag{6.15}$$

On the other hand, note that $2s + u \leq d|A_\tau|$, since each satisfied check in $\Gamma_1(A_\tau)$ must have at least two bits in x_{A_τ} to be as addends. As $A'_\tau = A_\tau \cap V$, we have $s' \leq s, u' \leq u$.

Thus

$$2s' + u' \leq 2s + u \leq d|A_\tau|. \tag{6.16}$$

Combining (6.15) and (6.16), we get

$$u' \geq 2(0.9d|A'_\tau| - 0.5d|A_\tau|). \tag{6.17}$$

Further by (6.14),(6.17),

$$u' \geq 0.68d|A_\tau| \geq 0.68d|A'_\tau|. \tag{6.18}$$

Hence by an averaging argument, there is an index $j \in V$ s.t. the number of unsatisfied parity checks in $\Gamma_1(j)$ is at least $0.68d$.

As a result, after doing the flipping for this round, the number of unsatisfied parity checks is strictly decreased. Also note that because of the restriction sets in our algorithm our operation cannot create an A_τ in some steps s.t. it does not have a good expansion. Hence, the first stage ends when $|A_\tau| < 2k'$.

Next we consider $|A_\tau| < 2k'$ at the beginning of a round τ. There are two possible cases.

The first case is that in step 2 the algorithm does not find a $j \in V$ to conduct the operation, so it will go to the next stage as desired.

The second case is that there is still an index $j \in V$ s.t. the number of unsatisfied parity checks in $\Gamma_1(j)$ is more than half. Hence after flipping, and the number of unsatisfied parity checks is again strictly decreased. Note that there are at most $O(m_1)$ unsatisfied checks. So this procedure will end in at most $O(m_1)$ rounds.

For either case, stage 1 will end with $|A_\tau| < 2k'$. This shows the claim. □

As a result, after stage 1, the number of errors is less than $2k'$.

As C_0 can correct $2k'$ errors, by Theorem 3.3, the decoding algorithm outputs x correctly. □

THEOREM 6.1. *There is an efficient one-way protocol for every $(\mathbf{s}, \mathbf{k}, t)$ DE, arbitrary $s_i = k2^{\Theta(i)}, l = \Omega(\log \frac{n}{k}), k_i = \max\{k/2^{\Theta(i)}, \Theta(\frac{k}{l \log \frac{n}{k}})\} \le s_i/40$, $t \le O(\sqrt{l})$, having communication complexity $O(k)$, success probability $1 - 2^{-\Theta(k \frac{\log\log \frac{n}{k}}{\log \frac{n}{k}})}$.*

Proof. The protocol is constructed by Lemma 6.2 and we will use a random (n, m, d) bipartite graph to be Γ_1. By Lemma 6.1, a random bipartite (n, m, d) graph Γ_1 is an $(n, m, d, \mathbf{S}, \overline{\mathbf{k}}, [k', k], 0.9)$ expander, with failure probability at most $\varepsilon = 2^{-\Theta(k' \log \frac{2k}{k'})}$, where we let $m = O(2dk), d = O(1)$, $\overline{k}_i = 20k_i, i \in [t]$, $k' = O(k/\log \frac{n}{k})$. Also since $t \le O(\sqrt{l})$, we have $k'' t \log \frac{2k2^t}{k''} \le k' \log \frac{2k}{k'}$, where $k'' = O(\frac{k}{l \log \frac{n}{k}})$.

By Lemma 6.3, Bob can compute x, by using $y, z, \mathbf{S}, \mathbf{k}, k'$ and the common randomness.

The communication complexity is $|z| = m = O(k)$. The protocol is efficient since both encoding and decoding are efficient. The failure probability is $\varepsilon = 2^{-\Theta(k \frac{\log\log \frac{n}{k}}{\log \frac{n}{k}})}$ since the construction of Γ_1 is the only part we use randomness. □

Note that Theorem 1.5 directly follows from Theorem 6.1 by letting $l = O(t^2)$.

7 Optimal Document Exchange under Edit Distance

In this section we give the one-way document exchange protocol for edit distance. We begin with a randomized protocol where the two parties have shared randomness.

CONSTRUCTION 7.1. *The input string for Alice has length $n \in \mathbb{N}$ and there are totally $k \in [\Theta(\log^4 \frac{n}{k}), \Theta(n)]$ edit errors between Alice's string and Bob's string.*

Both Alice's and Bob's algorithms have $L = O(\log \frac{n}{k})$ levels. For every $i \in [L]$, in the i-th level,

- *Let block size $b_i = \frac{n}{3 \cdot 2^i k}$, i.e., in each level we divide a block in the previous level evenly into two blocks; (We choose L properly s.t. $b_L = O(\log \frac{n}{k})$)*
- *The number of blocks $l_i = n/b_i$;*

Alice: On input $x \in \{0,1\}^n$;

1. *For the i-th level,*
   - 1.1. *Partition x into consecutive blocks $x[1, b_i], x[1+b_i, 2b_i], \ldots, x[1+(l-1)b_i, l_i b_i]$;*
   - 1.2. *Let $h_j : \{0,1\}^{b_i} \to \{0,1\}^c, j \in [l_i]$ be a sequence of random hash functions with c being a large enough constant positive integer;*
   - 1.3. *Compute $v[i][j] = h_j(x[1+(j-1)b_i, jb_i]), j \in [l_i]$;*
   - 1.4. *$v[i] = (v[i][1], \ldots, v[i][l_i])$;*
   - 1.5. *By the sketch construction of Theorem 6.1, compute $z[i] \in \{0,1\}^{m=O(k)}$, a sketch of $v[i]$, the expander constructed in this step being $\Gamma : \{0,1\}^{l_i} \times \{0,1\}^{d_1=10} \to \{0,1\}^m$;*
2. *Compute the redundancy $z_{\mathsf{final}} \in (\{0,1\}^{b_L})^{\Theta(k)}$ for the blocks of the L-th level by Theorem 3.3, where the code has distance $16k$;*
3. *Send $z = (z[1], z[2], \ldots, z[L])$, $v[1]$, z_{final}.*

Bob: On input $y \in \{0,1\}^{O(n)}$ and received z, $v[1]$, z_{final};

1. *Create $\tilde{x} \in \{0,1,*\}^n$ (i.e. his current version of Alice's x), initiating it to be $(*, *, \ldots, *)$;*
2. *Let $A_1 = [l_1], A_i = \emptyset, i = 2, 3, \ldots, L$;*
3. *For the i-th level, where $1 \le i \le L-1$,*
   - 3.1. *Divide $\tilde{x}$ into length b_i consecutive blocks, $\tilde{x}[1, b_i], \ldots, \tilde{x}[1+(l_i-1)b_i, l_i b_i]$;*
   - 3.2. *Utilize the common randomness to get functions $h_j : \{0,1\}^{b_i} \to \{0,1\}^c, j \in [l_i]$ that Alice gets in her stage 1.2.*
   - 3.3. *Compute $\tilde{v}[i] = (h_1(\tilde{x}[1, b_i]), \ldots, h_{l_i}(\tilde{x}[1+(l_i-1)b_i, l_i b_i]))$;*
   - 3.4. *For every $i' \in [\ell]$, let $S_{i'} \subseteq [l_i]$ be the indices of the (descendent) blocks in the current level, whose ancestors are those blocks indicated by $A_{i'}$, i.e. j is in $S_{i'}$ iff there is $j' \in A_{i'}$ s.t. $[1+(j-1)b_i, jb_i] \subseteq [1+(j'-1)b_{i'}, j'b_{i'}]$;*
   - 3.5. *Compute $v[i]$ by using the decoding algorithm from Construction 6.2 on input $\tilde{v}[i]$, $S_{i'}$, $k_{i'} = \max(k/2^{0.9c(i-i')}, k/\log^3 \frac{n}{k})$, $i' = i-1, i-2, \ldots, 1$, and the received $z[i]$;*

3.6. *Let $T_i = \emptyset$. For every $j \in [l_i]$, if $v[i][j] \neq \tilde{v}[i][j]$, then put $j \in T_i$ and then check every $i' = i-1, i-2, \ldots, 1$, if the j-th block in the current level is a descendent of the j'-th block in the i'-th level, then remove j' from $A_{i'}$;*

3.7. *Let $A_i = T_i$;*

3.8. *Compute $w \in (A_i \times [|y|])^{|w|}$ which is the maximum monotone matching between x's blocks indicated by A_i, and y, under $h_1, \ldots, h_{l_i}$, using $v[i]$, by Lemma 3.4; (We interpret w as a sequence of matches, the jth match being denoted as $(w[j][1], w[j][2])$.)*

3.9. *Evaluate $\tilde{x}$ according to w, i.e. let $\tilde{x}[w[j][1]] = y[w[j][2], w[j][2] + b_i - 1], \forall j \in [|w|]$;*

4. *In the L'th level, apply the decoding of Theorem 3.3 on the blocks of $\tilde{x}$ and z_{final} to get x;*

5. *Return x.*

Next we show the correctness of our construction.

Consider every level $i \in [L]$, every $i' = i-1, i-2, \ldots, 1$. We denote the set descendants in the i-th level, stemming from $A_{i'}$, as $\tilde{A}_{i'}$. The indices set of undetected wrongly recovered blocks in $\tilde{A}_{i'}$, is denoted as $B_{i'}$, $i' = i-1, \ldots, 1$.

Let i^* be s.t. $k'' \triangleq k'/\Theta(\log^2 \frac{n}{k}) \in [k/2^{c(i-i^*)}, k/2^{c(i-i^*+1)}]$, $k' \triangleq k/\log \frac{n}{k}$.

LEMMA 7.2. *For every $i \in [L]$, if $\forall i' < i, |T_{i'}| = O(k)$, and $v[i']$ are computed correctly by Bob, then*

- *for every $i' \leq i^*$, the probability that $|B_{i'}| \geq k''$ is at most $2^{-\Omega(k'')}$;*
- *for every $i' \in (i^*, i)$, the probability that $|B_{i'}| \geq k_{i'} = k/2^{0.9c(i-i')}$ is at most $2^{-\Omega(ck/2^{c(i-i')})}$.*

Proof. Consider the possibilities of $B_{i'}$. Each possibility can be described by a w-witness with $w = |B_{i'}|$. The witness is a sequence of (number) w indices where each index is in the i-th level indicating a wrongly recovered block. This sequence is further partitioned into $i-i'+1$ groups corresponding to levels $i', i'+1, \ldots, i$. We numerate these groups as group $i', i'+1, \ldots, i$.

Consider the trees rooted at blocks in $A_{i'}$. Each of them has height $i - i'$. Each node is a block in a certain level between i' and i.

The w-witness describes level i bad blocks which are descendants of blocks in $A_{i'}$, uniquely in the following way.

Group $j \in [i', i]$ consists of indices of bad blocks, one for each depth $i - j$ tree whose root is a wrong block in level j. Note that for one tree, there may be many bad leaf blocks. For this case, we only pick the leftmost wrong one. These forms the group one. After each picking, we cut all the edges from that block to the root. This gives $i - i' + 1$ sub-trees. One of them is the last block. We only focus on sub-trees other than that picked block. They have depth from 1 to $i - j$. We update the set of trees by adding these trees from cutting and delete the trees being cut.

In this way, every error patterns can be described. This is because, every leaf node is either being picked or still in one of the trees in the forest. Once the leaf is in one of the trees in the forest, it can be picked in a certain level of the picking procedure.

Let the number of wrong blocks being picked for each level j be w_j.

The total number of error patterns is

$$\begin{aligned} P =& \binom{k}{w_{i'}} \cdot 2^{(i-i')w_{i'}} \cdot \binom{w_{i'}}{w_{i'+1}} \cdot 2^{(i-i'-1)w_{i'+1}} \\ & \cdot \binom{w_{i'} + w_{i'+1}}{w_{i'+2}} \cdot 2^{(i-i'-2)w_{i'+2}} \cdots \binom{\sum_{j=i'}^{i-1} w_j}{w_i} \\ \leq & \binom{k}{w_{i'}} \binom{\sum_{j=i'}^{i-1}(i-j)w_j}{\sum_{j=i'}^{i} w_j} \cdot 2^{\sum_{j=i'}^{i-1}(i-j)w_j} \end{aligned}$$

For $i' \leq i^*$, suppose $\sum_{j=i'}^{i}(i-j)w_j = k''$. Then

$$\begin{aligned} P \leq & \binom{k}{k''/(i-i')} \cdot 2^{2k_0} \\ \leq & 2^{O(k'') \cdot \frac{O(\log k)}{i-i'}} \cdot 2^{2k''} \\ \leq & 2^{O(k'')}. \end{aligned} \tag{7.19}$$

Note that the probability that a specific error pattern happens is at most $2^{-c\sum_{j=i'}^{i}(i-j)w_j} = 2^{-ck''}$ because each block in group j is checked for $i - j$ times independently. Since c is a large enough constant, $\sum_{j=i'}^{i}(i-j)w_j$ is a integer in $[0, \mathsf{poly}(k \log n)]$, we know by a union bound, $\sum_{j=i'}^{i}(i-j)w_j \geq k''$ happens with probability at most $2^{-ck''} \times 2^{O(k'')} \times \mathsf{poly}(k \log n) \leq 2^{-\Omega(k'')}$.

For $i > i^*$, suppose $\sum_{j=i'}^{i}(i-j)w_j = k/2^{0.9c(i-i')} = k_{i'}$. Then

$$\begin{aligned} P \leq & \binom{k}{k_{i'}/(i-i')} \cdot 2^{2k_{i'}} \\ \leq & 2^{(0.9c(i-i') + O(1) + \log(i-i')) \cdot k_{i'}/(i-i')} \cdot 2^{2k_{i'}} \\ \leq & 2^{0.91ck_{i'}}, \end{aligned}$$

when c is a large enough constant.

Similarly, note that the probability that a specific error pattern happens is at most $2^{-c\sum_{j=i'}^{i}(i-j)w_j} = 2^{-ck_{i'}}$ because each block in group j is checked for $i-j$ times independently. Since c is a large enough constant, $\sum_{j=i'}^{i}(i-j)w_j$ is a integer in $[0, \mathsf{poly}(n)]$, we know by a union bound, $\sum_{j=i'}^{i}(i-j)w_j \geq k_{i'}$ happens with probability at most $2^{-ck_{i'}} \times 2^{0.91ck_{i'}} \times \mathsf{poly}(k\log n) \leq 2^{-\Omega(k_{i'})}$.

As a result, $w = \sum_{j=i'}^{i} w_j > k_{i'}$ happens with probability at most $2^{-\Omega(k_{i'})} \leq 2^{-\Omega(k'')}$. ∎

LEMMA 7.3. *For every $i \in [L]$, if $\forall i' < i, |T_{i'}| = O(k)$, and $v[i']$ are computed correctly by Bob, then with probability at least $1 - 2^{-\Omega(k'')}$,*

$$\sum_{i'=1}^{i-1} |B_{i'}| < k.$$

Proof. By Lemma 7.2, for $i' \leq i^*$, with probability at least $1-2^{-\Omega(k'')}$, $|B_{i'}| < k''$; for $i' > i^*$, with probability at least $1 - 2^{-\Omega(k'')}$, $|B_{i'}| \leq k_{i'} = k/2^{0.9c(i-i')}$.

By a union bound, with probability at least $1 - i2^{-\Omega(k'')} = 1 - 2^{-\Omega(k'')}$,

$$\begin{aligned}\sum_{i'=1}^{i-1} |B_{i'}| =& \sum_{i'=1}^{i^*} |B_{i'}| + \sum_{i'=i^*+1}^{i-1} |B_{i'}| \\ \leq& (i^*-1)k'' + 0.5k < k.\end{aligned}$$

∎

LEMMA 7.4. *For every $i \in L$, at level i, if $v[i]$ are computed correctly by Bob, and $|T_i| \leq 6k$, then with probability $1 - 2^{-\Theta(k)}$, the number of wrongly recovered blocks introduced by w is at most k.*

Proof. Assume the number of wrongly recovered blocks introduced by w is larger than k. Then there more than k pairs in the matching are bad pairs. This happens with probability $1/2^{ck}$.

Note that by Lemma 3.4, for w, $|\rho'_1 - \rho_1| + |(\rho'_2 - \rho'_1) - (\rho_2 - \rho_1)| + \cdots + |(\rho'_{|w|} - \rho'_{|w|-1}) - (\rho_{|w|} - \rho_{|w|-1})| \leq k$. By Lemma 3.3, since $|T_i| \leq 6k$, there are totally $2^{O(k)}$ possible matchings that can be output by our algorithm.

So by a union bound, the conclusion holds with probability $1 - 2^{-\Theta(k)}$. ∎

LEMMA 7.5. *For every $i \in L$, in level i, if $v[i]$ are computed correctly, and $|T_i| = O(k)$, then with probability $1 - 2^{-\Theta(k)}$, the number of wrongly recovered blocks and uncovered blocks in T_i after 3.9. is at most $2k$.*

Proof. By Lemma 3.4, $|w| \geq |T_i| - k$. Thus the number of uncovered blocks is at most k. By Lemma 7.4, with probability $1 - 1/2^{\Theta(k)}$, the number of wrongly recovered blocks introduced by w is at most k. So the total number of wrongly recovered blocks is at most $2k$. ∎

LEMMA 7.6. *For every $i \in L$, with probability $1 - 2^{-\Theta(k'')}$,*

- *after the first step of level i, the number of wrongly recovered blocks is at most $6k$;*
- *Bob can compute $v[i]$ correctly;*
- *the number of wrongly recovered blocks in T_i is at most $2k$ after step 3.9..*

Proof. We use induction.

In the first level, $\tilde{x} = (*, *, \ldots, *)$. So the number of wrongly recovered blocks at the beginning is $l_1 = n/b_1 = 6k$. So The number of wrongly recovered blocks is at most $6k$. Also Bob can get $v[1]$ correctly, since it is directly sent by Alice. By Lemma 7.5, with probability $1-1/2^{\Theta(k)}$, the total number of wrongly recovered blocks is at most $2k$ if we regard uncovered blocks as wrongly recovered.

Suppose the conclusion holds for the first $i-1$-level. Consider level i.

By Lemma 7.3, with probability $1 - 2^{-\Omega(k'')}$, the total number of wrongly recovered blocks is $\sum_{i'=1}^{i-1} |B_{i'}| < k$.

By Lemma 6.1, with probability $1 - \varepsilon_1 = 1 - 2^{-\Omega(k')}$, Γ_1 is a bipartite graph, having $n_1 = l_i$ left vertices, $m = O(k)$ right vertices, left degree $d = O(1)$, s.t. $\forall R \subseteq [n_1], |R| \in [k', k], |R \cap S_{i'}| \leq k'_{i'} = \max(20k/2^{c(i-i')}, 20k^{0.9})$,

$$\Gamma(R) > 0.9d|R|.$$

Note that $k'_{i'} \geq 20k_{i'}$. Also note that i' iterates in $[1, i-1]$. So the number of $S_{i'}$ is at most $L \leq k^{\beta/2}\sqrt{\log k}$. So by Theorem 6.1, Bob can get the correct $v[i]$.

As a result, by a union bound with probability $1 - L2^{-\Theta(k'')}$, Bob can compute $v[i]$ correctly. Note that $L = O(\log \frac{n}{k})$, $k = \Omega(\log^4 \frac{n}{k})$. So the probability is at least $1 - 2^{-\Theta(k'')}$.

By Lemma 7.5, with probability $1 - 1/2^{\Theta(k)}$, the total number of wrongly recovered blocks in T_i is at most $2k$ after stage 3.9..

So the overall probability is as desired.

This shows the inductive step. ∎

LEMMA 7.7. *With probability $1 - 2^{-\Theta(k'')}$, Bob outputs x correctly.*

Proof. By Lemma 7.6, with probability $1 - 2^{-\Theta(k'')}$, at the last level, there are at most $6k$ wrong blocks. Since z_{final} is the redundancy for a code with distance $16k$, all wrong blocks can be corrected. So Bob computes x correctly. ∎

LEMMA 7.8. *The communication complexity is* $O(k \log \frac{n}{k})$.

The proof is deferred to the full version.

THEOREM 7.1. *There exists an efficient one-way edit distance document exchange protocol using common randomness, for every* $n \in \mathbb{N}$, $k = \Omega(\log^4 \frac{n}{k})$, *having sketch length* $O(k \log \frac{n}{k})$, *success probability* $1 - 2^{-\Omega(k/\log^3 \frac{n}{k})}$.

Proof. It immediately follows from Lemma 7.7, 7.8. The protocol is efficient since all components and steps are efficient. ∎

By combining Theorem 7.1 and the result of Haeupler [20], we immediately get the following.

THEOREM 7.2. *There exists an efficient one-way edit distance document exchange protocol using common randomness, for every* $n, k \in \mathbb{N}$, *having sketch length* $O(k \log \frac{n}{k})$, *success probability* $1 - \min\{2^{-\Theta(k/\log^3 \frac{n}{k})}, 1/\mathsf{poly}(n)\}$.

7.1 Removing Shared Randomness In Construction 7.1, we use common randomness to generate hash functions $h_j, j \in [l_i]$ for each $i \in [L]$. Also we use common randomness to generate the random bipartite graph Γ for the encoding of the hash values. Now we show that we can use almost κ-wise independence generator to reduce randomness.

LEMMA 7.9. *Replace the common randomness used in Construction 7.1,*

- *for generating hash functions, by an* ϵ*-almost* $10ck$*-wise independent distribution, with* $\epsilon = 2^{-10ck}$;
- *for generating* Γ_1, *by* $O(k)$*-wise independent distributions over alphabet* $[m]$. *(Recall that* $m = O(k)$*) Then with probability* $1 - 2^{-\Theta(k')}$, *Bob outputs* x *correctly.*

The proof is deferred to the full version.

THEOREM 7.3. *There exists an efficient one-way edit distance document exchange protocol, for every* $k = \Omega(\log^4 \frac{n}{k})$, *having sketch length* $O(k \max\{\log \frac{n}{k}, \log k\})$, *success probability* $1 - 2^{-\Omega(k/\log^3 \frac{n}{k})}$.

The proof is deferred to the full version.

References

[1] Khaled A. S. Abdel-Ghaffar and Amr El Abbadi. An optimal strategy for comparing file copies. *IEEE Transactions on Parallel and Distributed Systems*, 5(1):87–93, 1994.

[2] Micah Adler, Erik D. Demaine, Nicholas J.A. Harvey, and Mihai P?atra? Lower bounds for asymmetric communication channels and distributed source cod. In *SODA*, pages 251–260, 2006.

[3] Micah Adler and Bruce M Maggs. Protocols for asymmetric communication channels. *Journal of Computer and System Sciences*, 63(4):573–596, 2001.

[4] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures & Algorithms*, 3(3):289–304, 1992.

[5] Alexandr Andoni, Javad Ghaderi, Daniel Hsu, Dan Rubenstein, and Omri Weinstein. Coding sets with asymmetric information. *ArXiv e-prints*, 2018.

[6] Daniel Barbara and Hector Garcia-Molina. Exploiting symmetries for low-cost comparison of file copies. In *[1988] Proceedings. The 8th International Conference on Distributed*, pages 471–479. IEEE, 1988.

[7] Daniel Barbara and Richard J. Lipton. A class of randomized strategies for low-cost comparison of file copies. *IEEE Transactions on Parallel and Distributed Systems*, 2(2):160–170, 1991.

[8] Djamal Belazzougui and Qin Zhang. Edit distance: Sketching, streaming, and document exchange. In *Proceedings of the 57th IEEE Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2016.

[9] Raj Chandra Bose and Dwijendra K Ray-Chaudhuri. Further results on error correcting binary group codes. *Information and Control*, 3(3):279–290, 1960.

[10] Boris Bukh and Venkatesan Guruswami. An improved bound on the fraction of correctable deletions. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1893–1901. ACM, 2016.

[11] Diptarka Chakraborty, Elazar Goldenberg, and Michal Koucký. Low distortion embedding from edit to hamming distance using coupling. In *Proceedings of the 48th IEEE Annual Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2016.

[12] Kuan Cheng, Zhengzhong Jin, Xin Li, and Ke Wu. Deterministic document exchange protocols, and almost optimal binary codes for edit errors. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 200–211. IEEE, 2018.

[13] Kuan Cheng, Zhengzhong Jin, Xin Li, and Ke Wu. Block edit errors with transpositions: Deterministic document exchange protocols and almost optimal binary codes. In *46th International Colloquium on Automata,*

Languages, and Programming (ICALP 2019). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[14] Kuan Cheng and Xin Li. Efficient document exchange and error correcting codes with asymmetric information. *arXiv preprint arXiv:2007.00870*, 2020.

[15] Graham Cormode, Mike Paterson, Suleyman Cenk Sahinalp, and Uzi Vishkin. Communication complexity of document exchange. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 197–206. ACM, 2000.

[16] Arnaldo Garcia and Henning Stichtenoth. On the asymptotic behaviour of some towers of function fields over finite fields. *Journal of number theory*, 61(2):248–273, 1996.

[17] V. Guruswami and R. Li. Efficiently decodable insertion/deletion codes for high-noise and high-rate regimes. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 620–624, July 2016.

[18] V. Guruswami and C. Wang. Deletion codes in the high-noise and high-rate regimes. *IEEE Transactions on Information Theory*, 63(4):1961–1970, April 2017.

[19] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. *Journal of the ACM*, 56(4), 2009.

[20] Bernhard Haeupler. An optimal document exchange protocol. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, 2019.

[21] Bernhard Haeupler and Amirbehshad Shahrasbi. Synchronization strings: codes for insertions and deletions approaching the singleton bound. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 33–46. ACM, 2017.

[22] Bernhard Haeupler and Amirbehshad Shahrasbi. Synchronization strings: Explicit constructions, local decoding, and applications. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing*, 2018.

[23] Alexis Hocquenghem. Codes correcteurs derreurs. *Chiffres*, 2(2):147–56, 1959.

[24] Tom Høholdt, Jacobus H Van Lint, and Ruud Pellikaan. Algebraic geometry codes. *Handbook of coding theory*, 1(Part 1):871–961, 1998.

[25] Utku Irmak, Svilen Mihaylov, and Torsten Suel. Improved single-round protocols for remote file synchronization. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1665–1676. IEEE, 2005.

[26] Hossein Jowhari. Efficient communication protocols for deciding edit distance. In *ESA*, 2012.

[27] Eduardo Sany Laber and Leonardo Gomes Holanda. A new protocol for asymmetric communication channels: Reaching the lower bounds. *Scientia Iranica*, 8(4):297–302, 2001.

[28] Eduardo Sany Laber and Leonardo Gomes Holanda. Improved bounds for asymmetric communication protocols. *Information Processing Letters*, 83(4):205–209, 2002.

[29] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, February 1966.

[30] A Orlitsky and K Viswanathan. Practical algorithms for interactive communication. In *IEEE Int. Symp. on Information Theory*, 2001.

[31] Alon Orlitsky. Worst-case interactive communication 1: Two messages are almost optimal. *IEEE transactions on Information Theory*, 36:1111–1126, 1990.

[32] Alon Orlitsky. Interactive communication: Balanced distributions, correlated files, and average-case complexity. In *[1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science*, pages 228–238. IEEE, 1991.

[33] L. J. Schulman and D. Zuckerman. Asymptotically good codes correcting insertions, deletions, and transpositions. *IEEE Transactions on Information Theory*, 45(7):2552–2557, Nov 1999.

[34] Kenneth W Shum, Ilia Aleshnikov, P Vijay Kumar, Henning Stichtenoth, and Vinay Deolalikar. A low-complexity algorithm for the construction of algebraic-geometric codes better than the gilbert-varshamov bound. *IEEE Transactions on Information Theory*, 47(6):2225–2241, 2001.

[35] Michael Sipser and Daniel A Spielman. Expander codes. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 566–576. IEEE, 1994.

[36] Michael Sipser and Daniel A Spielman. Expander codes. *IEEE transactions on Information Theory*, 42(6):1710–1722, 1996.

[37] David Slepian and Jack Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on information Theory*, 19(4):471–480, 1973.

[38] Torsten Suel, Patrick Noel, and Dimitre Trendafilov. Improved file synchronization techniques for maintaining large replicated collections over slow networks. In *Proceedings. 20th International Conference on Data Engineering*, pages 153–164. IEEE, 2004.

[39] John Watkinson, Micah Adler, and Faith E Fich. New protocols for asymmetric communication channels. In *SIROCCO*, 2001.