Budget Error-Correcting under Earth-Mover Distance

Christian Konrad[1], Wei Yu[2], and Qin Zhang[3]

[1] LIAFA, Université Paris Diderot, France
konrad@lri.fr
[2] Aarhus University
yuwei@cs.au.dk
[3] IBM Almaden
qinzhang@cs.au.dk

Abstract. We study the following budget error-correcting problem: Alice has a point set x and Bob has a point set y in the d-dimensional grid. Alice wants to send a short message to Bob so that Bob can use this information to adjust his point set y towards x to minimize the Earth-Mover Distance between the two point sets. A more intuitive way to understand this problem is: Alice tries to help Bob to recall Eve's face by sending him a short message. Of course Bob will fail to recall if he does not know Eve, but if he knows something about Eve, the message could help a lot.

Naturally, there is a trade-off between the message size and the quality of such an adjustment. Now given a quality constraint, we want to minimize the message size. This problem is well motivated by applications including image exchange/synchronization and video compression. In this paper, we give almost matching upper and lower bounds for this problem.

1 Introduction

In this paper we study the following two-party one-way communication problem.

Definition 1 (The EMD k-Budget Error-Correcting). *We have two players Alice and Bob. Alice is given a set of n points $x = \{x_1, \ldots, x_n\} \subseteq [\Delta]^d$ on the d-dimensional grid $[\Delta]^d$, and Bob is given another set of n points $y = \{y_1, \ldots, y_n\} \subseteq [\Delta]^d$. Alice sends a message M to Bob. Bob then tries to relocate his points to $y^* = \{y_1^*, \ldots, y_n^*\} \subseteq [\Delta]^d$ such that*

$$\mathrm{EMD}(x, y^*) \leq C \min_{\tilde{y} \in \mathcal{N}_k(y)} \mathrm{EMD}(x, \tilde{y}),$$

where the Earth-Mover Distance (EMD) between two point sets x, y of size n is defined as the minimum perfect matching between x, y, that is,

$$\mathrm{EMD}(x, y) = \min_{\pi:[n]\to[n]} \sum_{1 \leq i \leq n} \left\| x_i - y_{\pi(i)} \right\|_2 .$$

$\mathcal{N}_k(y)$ denotes all point sets of cardinality n that can be obtained by relocating k points in y, and C is a fixed approximation factor. The goal is to minimize the message size $|M|$.

1.1 Motivations

This work is largely motivated by the following general problem in the two-party one-way communication. We have two players Alice and Bob. Alice holds an object x and Bob holds an object y. Alice wants to send Bob a message so that either Bob can learn Alice's input x, or he can report that x and y are far apart under a certain measurement. The goal is to minimize the message size.

As a concrete example, consider the classic *document exchange* problem [6, 16, 18]. In this problem Alice has a string x and Bob has a string y. Alice wants to send a short message to Bob so that either Bob can learn x or he can assert that the Edit Distance[4] between x and y is at least k. The reason of introducing a parameter k into the problem can be seen from the following application: Suppose that Alice wants to send a large file to Bob through a noisy channel. She can supplement the file with a short sketch M of the file so that if the number of errors introduced in the transmission is small, then Bob can correct them using M without any further communication[5]. Otherwise, which is often quite unlikely, Bob can detect that the file is heavily corrupted. In this case he can simply ask Alice to retransmit the whole file. Such a strategy can greatly reduce the cost of error correction.

This general problem is also quite useful in the object (e.g., document, image, etc.) synchronization, where it is natural to assume that the similarity between Alice's object x and Bob's object y is high, therefore a short sketch of x can let Bob recover x.

The strategy that either we correct at most k mistakes without additional communication or we retransmit everything works fine for some distance measurements, e.g., the Hamming Distance and the Edit Distance. However, it is not suitable for measurements like the Earth-Mover Distance (EMD). EMD is a popular distance function to compare the similarity between two images in the field of computer vision (see, e.g., [22, 11, 10, 4, 21]). During the transmission of an image, it is possible that the whole image is slightly shifted or rotated. Such a noise often will not affect the usefulness of the image, but according to the previous error-correcting strategy, the whole image has to be retransmitted since the EMD is high between the original image and the one after the shift/rotation.

To give a better solution to this scenario, we borrow an idea from the compressive-sensing/sparse-recovery literature (see [9] for a survey). In typical settings of sparse recovery, Alice has an n-dimensional vector x. She sends Bob a linear sketch Ax where A is an $m \times n$ $(m \ll n)$ matrix, and then Bob tries to reconstruct a vector x^* such that

$$\|x - x^*\|_p \leq C \min_{k\text{-sparse } \tilde{x}} \|x - \tilde{x}\|_q$$

where p, q are norm parameters, $C > 0$ is the approximation factor, and we say a vector x is k-sparse if it has at most k non-zero coordinates. The primary goal is to minimize m, that is, the number of rows of the sketch matrix A. A sparse-recovery scheme can potentially be used for our error-correcting setting: Alice computes Ax and

[4] That is, the minimum number of character insertions/deletions/substitutions needed to convert x to y.

[5] We can assume that M is transmitted via an error-correcting code so that Bob can learn M without error. This will only increase the message size by a constant factor.

sends the result to Bob. Bob then computes Ay and $(Ax - Ay) = A(x - y)$. He then tries to recover the important coordinates of $(x - y)$ from $A(x - y)$. One issue is that now $(x - y)$ contains negative coordinates. Indyk and Price [15] studied the sparse recovery for $p = q = \text{EMD}$, but the problem they formulated is different from ours (note that x^* being k-sparse is different from $x^* \in \mathcal{N}_k(\mathbf{0})$). Our problem is more like an "approximate" error-correction code. Also, their algorithm only works for vectors with positive coordinates thus cannot be used to estimate $(x - y)$. Another difference is that in our problem we do not restrict Alice's message to be a linear sketch.

Besides error correction and object synchronization, our problem also has other potential applications. For example, in the image/video compression, consider the scenario that we use a sensor to monitor a herd of cows on a farm. In sensor networks we would like to minimize the communication since the transmission of data is the biggest energy drain. It is reasonable to assume that in most consecutive time steps only a few cows move a large distance from their original positions, while the others stay in a small neighbourhood. Therefore in order to save communication, instead of sending a new image at each time step, the sensor can only send the central server a short message which is enough for the server to detect/recover the positions of those cows whose new positions deviate significantly from their original positions. A similar idea has already been used in [24] for smart cameras (e.g. phone cameras) for detecting moving objects.

1.2 Results and Techniques

In this paper we show the following results for EMD k-Budget Error-Correcting.

- We give an $O(d)$-approximation randomized protocol with $\tilde{O}(k \log \Delta \log(n\Delta^d))$ bits[6] of communication.
- We complement our upper bound with a lower bound of $\Omega(k \log \Delta \log(\Delta^d / k)/\log d)$ bits of communication. The lower bound holds for randomized algorithms that compute an $O(d)$-approximation.

Note that for typical settings where $d = O(1), n = \Delta^{O(1)}$, the upper bound almost matches the lower bound. The stated communication complexity protocol can actually be implemented in polynomial time. See Section 2 for details.

Below we summarize the general ideas of our approaches.

Upper bound. We illustrate our algorithm in the one dimensional case. Given Alice's input x and Bob's input y on the one dimensional grid $[\Delta]$, the optimal solution will return a set of k pairs of points $\{(u_1, v_1), \ldots, (u_k, v_k)\}$ $(u_i \in x, v_i \in y)$ so that if Bob moves point v_i to u_i for all $i \in [k]$, the EMD between Alice's point set x and Bob's modified point set $y_{\texttt{OPT}}$ is minimized. Intuitively, we can view the k pairs of points as the top k edges of a perfect matching between x and y, and a good algorithm will try to report those edges.

W.l.o.g, we assume that $\Delta = 2^L$ for some integer L. Firstly, we employ an idea that was already used in [14]. Alice builds a hierarchical partition (we later call it *pyramid arrays*) $\text{PA}(x) = \{\text{PA}_0(x), \text{PA}_1(x), \ldots, \text{PA}_L(x)\}$ for x, where $\text{PA}_i(x)$ is an array

[6] We use $\tilde{O}(f)$ to denote a function of the form $O(f \log f)$.

containing 2^i elements, and the j-th element of $\mathrm{PA}_i(x)$ contains the number of points in x that fall into the interval $\big((j-1)\cdot 2^{L-i}, j\cdot 2^{L-i}\big]$. Bob builds a similar hierarchical partition $\mathrm{PA}(y)$ for y. It is easy to see that for two points u, v ($u \in x, v \in y$) such that $2^{L-r} \le dist(u,v) < 2^{L-r+1}$, u and v must lie in cells with different indices in $\mathrm{PA}_\ell(x)$ and $\mathrm{PA}_\ell(y)$ for all $r \le \ell \le L$, thus (u,v) will contribute 2 to $\|\mathrm{PA}_\ell(x) - \mathrm{PA}_\ell(y)\|_1$ for $r \le \ell \le L$. On the other hand, u and v will very likely lie in cells with the same index in $\mathrm{PA}_\ell(x)$ and $\mathrm{PA}_\ell(y)$ for all $0 \le \ell < r$, thus (u,v) will not contribute to $\|\mathrm{PA}_\ell(x) - \mathrm{PA}_\ell(y)\|_1$ for $0 \le \ell < r$. Therefore a natural idea is to first find the largest ℓ such that $\|\mathrm{PA}_\ell(x) - \mathrm{PA}_\ell(y)\|_1 \le 2\alpha k$ for some small constant $\alpha > 1$, and then Alice sends an encoding of $\mathrm{PA}_\ell(x)$ to Bob, from which Bob can compute a set of edges possibly including the k longest ones. There are two issues we need to consider.

1. It is possible that for a pair of points (u,v) with $dist(u,v) < 2^{L-r}$, u and v lie in cells with different indices in $\mathrm{PA}_\ell(x)$ and $\mathrm{PA}_\ell(y)$ for an $\ell \le r$. In this case we may introduce a "false positive" in the relocation.
2. For a pair (u,v) ($u \in x, v \in y$), Bob can only learn from $\mathrm{PA}_\ell(x)$ that a point u lies in some interval $\big((j-1)\cdot 2^{L-\ell}, j\cdot 2^{L-\ell}\big]$, and he still does not know the exact location of u to where he should relocate his corresponding point v.

To handle the first issue, we simply perform a random shift of all points in x and y. In doing so, we can guarantee with a good probability that there are not many false positives. Such random shifts were used before, e.g., in [13]. To handle the second issue, we introduce a constant redundancy factor $\alpha > 1$ in the algorithm. That is, Alice sends a message to Bob so that Bob is able to relocate αk ($> k$) points (if needed). Now in the case when Bob cannot decide the exact location that he should relocate a point but only an interval, he simply moves the point to an arbitrary location on that interval. Such an operation will introduce an additional error, but since the optimal solution can only relocate k points, we can charge the errors that we make when relocating each point to the error that the optimal solution has to make on the (at least) $(\alpha-1)k$ points that it is unable to relocate. Some extra difficulties come from the interplay of these two issues. For example, it is possible that the relocation of a false positive will again introduces some error. Thus we need to carefully design the charging scheme so that such errors can also be charged to the error that the optimal solution will make.

Remark 1. To the best of our knowledge, all previous works on computing EMD-related problems using a hierarchical partition plus a random shift only give a logarithmic approximation, given certain polylogarithmic space/communication constraints. For example, even if Bob had to estimate $\mathrm{EMD}(x,y)$, there is no polylogarithmic communication protocol that would achieve a constant approximation in our one-way communication setting. The best known uses polynomial space [1, 23]. Thus it is very interesting to the authors that computing the "residual EMD distance" (i.e., EMD k-Budget Error-Correcting) actually admits a constant approximation (when the dimension is a constant).

Lower bound. We again illustrate the idea through the one dimensional case. We first describe a family of hard instances for EMD k-Budget Error-Correcting on the one dimensional grid $[\varDelta]$. Alice and Bob hold sets of n points x and respectively y on grid

$[\mathit{\Delta}]$. The construction is performed in two steps. In the first step, we choose p *point center* locations $1, \mathit{\Delta}/p + 1, 2\mathit{\Delta}/p + 1, \ldots, (p-1)\mathit{\Delta}/p + 1$, and in both x and y we assign n/p points to each point center. In the second step, we move points from these point centers in x and y to the right. At this step we make the point sets x and y different. We pick L $(= \Theta(\log \mathit{\Delta}))$ subsets $X_1, \ldots, X_L \subseteq [p]$ such that $|X_i| = k$ for all $i \in [L]$. In x, for all $i \in [L]$, for all $j \in X_i$, we move one point in the j-th point center by a distance of 2^{Bi} where B is a technical parameter. In y we perform similar operations: we first pick a random $I \in [L]$, and then for all $i = \{I+1, \ldots, L\}$, for all $j \in X_i$, we move one point from the j-th point center by a distance of 2^{Bi}. Note that x and y differ by those points that are moved in x indicated by $X_1, \ldots, X_I$. These points remain in point centers in y. The k most significant differences in point set x and y are the k points that Alice moved by distance 2^{BI}, that is, those points indicated by X_I. Intuitively, if Bob wants to correct most of these points, Bob has to learn X_I approximately.

The technical implementation of this idea is a reduction from the well-known two-party one-way communication problem called Augmented Indexing. Augmented Indexing has been used to prove lower bounds in streaming and sparse-recovery literature [5, 19, 7]. In Augmented Indexing, Alice has $(X_1, \ldots, X_L)$ and Bob has $(X_{I+1}, \ldots, X_L)$ for some index $I \in [L]$. Alice sends a single message to Bob and upon reception Bob outputs X_I. This problem is hard in the sense that since Alice does not know I, she has to send many of X_i $(i = 1, \ldots, L)$ to Bob so that Bob can output X_I correctly. In our application, each X_i $(i \in [L])$ is a subset of $[p]$ of cardinality k. The main difficulty lies in the fact that we aim to solve Augmented Indexing given a protocol for EMD k-Budget Error-Correcting that only computes a constant factor approximation. The key of our argument is that on our hard instances a constant factor approximation to EMD k-Budget Error-Correcting must identify *many* of the k points indicated by X_i. We use a family of k-subsets with bounded intersection which is similar to a binary constant weight code such that those identified points are enough to recover the correct k-subset, that is X_i. We comment that similar ideas have also been used in [7] for proving lower bounds for sparse-recovery problems.

2 The Upper Bound

Given two arrays A, B of the same length, we define $\|A - B\|_1 = \sum_i |A[i] - B[i]|$. For the upper bound we need an encoding scheme for arrays with certain properties. This encoding is stated in Lemma 1. Due to space constraints, the proof of the lemma is deferred to Appendix A.

Lemma 1. *Let x and y be two arrays of length N each. Each element in x and y is from $\{0, \ldots, n\}$. x could be encoded in $O(\log(1/\epsilon)\log(nN) + t\log(nN))$ bits, with which one having y in hand can*

- *fully decode x when $\|x - y\|_1 \le t$; or*
- *output "impossible" when $\|x - y\|_1 > t$.*

with probability $1 - \epsilon$ in encoding time $O(N^2 \log^2 n)$ and decoding time $O(N^2 \log^2 n)$.

We need the definition of pyramid arrays for the upper bound.

1. Alice randomly shifts all her points by a distance vector $\delta > 0$. More precisely, Alice first picks a d-dimensional vector $\delta \in [\Delta]^d$ uniformly at random, and then shifts all her points by δ. Let $x' = \{x'_{\mathbf{i}} \mid \mathbf{i} \in [2\Delta]^d\}$ be the characteristic d-dimensional vector of Alice's point set after the shift, that is, $x'_{\mathbf{i}}$ ($\mathbf{i} \in [2\Delta]^d$) counts the number of Alice's points at location $\mathbf{i}$. Alice includes δ in her message to Bob. Bob does the same operation with the same δ to his point set and gets a vector $y' = \{y'_{\mathbf{i}} \mid \mathbf{i} \in [2\Delta]^d\}$.
2. Alice and Bob construct d dimensional pyramid arrays $\mathrm{PA}^d(x')$ and $\mathrm{PA}^d(y')$ of x' and y', respectively.
3. For each level $\ell = 0, 1, \ldots, L = \log(2\Delta)$, Alice sends Bob a message M_ℓ according to Lemma 1 so that Bob can distinguish whether $\left\|\mathrm{PA}^d_\ell(x') - \mathrm{PA}^d_\ell(y')\right\|_1 > 2\alpha k$ or $\left\|\mathrm{PA}^d_\ell(x') - \mathrm{PA}^d_\ell(y')\right\|_1 \leq 2\alpha k$. Bob then finds the largest level ℓ^* such that $\left\|\mathrm{PA}^d_{\ell^*}(x') - \mathrm{PA}^d_{\ell^*}(y')\right\|_1 \leq 2\alpha k$ while $\left\|\mathrm{PA}^d_{\ell^*+1}(x') - \mathrm{PA}^d_{\ell^*+1}(y')\right\|_1 > 2\alpha k$, and reconstructs $\mathrm{PA}^d_{\ell^*}(x')$ according to Alice's message for level ℓ^*.
4. Bob picks an arbitrary vector $z' = \{z'_{\mathbf{i}} \mid \mathbf{i} \in [2\Delta]^d\}$ with the constraint that the ℓ^*-th level of the pyramid arrays of z', that is, $\mathrm{PA}^d_{\ell^*}(z')$, is equal to $\mathrm{PA}^d_{\ell^*}(x') - \mathrm{PA}^d_{\ell^*}(y')$. Let $z = \{z_{1,\ldots,1}, \ldots, z_{\Delta,\ldots,\Delta}\}$ where $z_{\mathbf{i}} = z'_{\mathbf{i}+\delta}$ for all $\mathbf{i} \in [\Delta]^d$. Finally, Bob relocates his n input points on grid $[\Delta]$ so that the corresponding characteristic vector changes from y to $y^* = y + z$.

Note that the whole message Alice sends to Bob is $M = \{\delta, M_0, M_1, \ldots, M_L\}$.

Fig. 1. Algorithm for EMD k-Budget Error-Correcting in d dimension.

Definition 2 **(d-Dimensional Pyramid Arrays).** *Let $x = \{x_{\mathbf{i}} \mid \mathbf{i} \in [\Delta]^d\}$ be a vector of length Δ^d. W.l.o.g., assume that $\Delta = 2^L$ for some integer L. We define the* d-dimensional pyramid arrays *of x to be a set of arrays $\mathrm{PA}^d(x) = \{\mathrm{PA}^d_0(x), \ldots, \mathrm{PA}^d_L(x)\}$, where $\mathrm{PA}^d_i(x)$'s are constructed as follows.*

1. $\mathrm{PA}^d_\ell(x) = \{a_{\ell,\mathbf{i}} \mid \mathbf{i} \in [2^\ell]^d\}$ is an array of size $2^{d\ell}$, for each $\ell \in \{0, 1, \ldots, L\}$.

2. $\mathrm{PA}^d_L(x) = \{a_{L,\mathbf{i}} \mid \mathbf{i} \in [2^L]^d\}$ is simply constructed by assigning $a_{L,\mathbf{i}} = x_{\mathbf{i}}$ for each $\mathbf{i} \in [2^L]^d (= [\Delta]^d)$.

3. For $\ell = 0, 1, \ldots, L-1$, $\mathrm{PA}^d_\ell(x) = \{a_{\ell,\mathbf{i}} \mid \mathbf{i} \in [2^\ell]^d\}$ is constructed by assigning $a_{\ell,\mathbf{i}} = \sum_{s \in \{0,1\}^d} a_{\ell+1,2\mathbf{i}+s}$ for each $\mathbf{i} \in [2^\ell]^d$.

The d-dimensional pyramid arrays can naturally be seen as a tree with the coordinates of $\mathrm{PA}^d_L(x)$ as leaves. Each coordinate r of $\mathrm{PA}^d_\ell(x)$ $(0 \leq \ell \leq L-1)$ corresponds to an internal node of the tree, and the value of coordinate r is the sum of the values of all leaves in the subtree rooted at r. For a leaf u, let $r_\ell(u)$ be the internal node at level ℓ such that u is in the subtree rooted at node $r_\ell(u)$.

Theorem 1. *There exists an algorithm that gives an $O(d)$-approximation for the EMD k-Budget Error-Correcting problem on d-dimensional grid $[\Delta]^d$ with communication $\tilde{O}(k \log \Delta \log(n\Delta^d))$ bits and success probability $2/3$.*

Set $\alpha = 6$ and $\beta = 2$. Let the approximation ratio be $C = 10d$ [7]. The algorithm is depicted in Figure 1. Now we show its correctness and analyze its communication complexity.

Correctness. Let $y_{\texttt{OPT}} \in \mathcal{N}_k(y)$ be such that $\mathrm{EMD}(x, y_{\texttt{OPT}})$ is minimized. Our goal is to show that $\mathrm{EMD}(x, y^*) \le C \cdot \mathrm{EMD}(x, y_{\texttt{OPT}})$ with probability $2/3$.

Let $\{(u_1, v_1), \ldots, (u_k, v_k)\}$ be a set of pairs such that in the optimal solution Bob moves one of his points at location v_i to u_i (that is, to match one of Alice's point at location u_i) for each $i \in [k]$. Let $(u_{k+1}, v_{k+1}), \ldots, (u_n, v_n)$ be a minimum perfect matching between the rest $(n-k)$ of Bob's points with the rest $(n-k)$ of Alice's points. If there are more than one such minimum perfect matchings, the choice can be made arbitrarily. W.l.o.g., we assume that $\|u_1 - v_1\|_2 \ge \|u_2 - v_2\|_2 \ge \ldots \ge \|u_n - v_n\|_2$.

Now let's focus on the level ℓ^* computed by Bob. Recall that ℓ^* is the largest level $\ell \in [L] \cup 0$ such that $\left\|\mathrm{PA}_\ell^d(x') - \mathrm{PA}_\ell^d(y')\right\|_1 \le 2\alpha k$. Let $A = \sqrt{d} \cdot 2^{L-\ell^*}$ be the diagonal distance of a grid cell in level ℓ^*. Obviously, for all those pairs (u_i, v_i) $(i \in [n])$ with $\|u_i - v_i\|_2 \ge A$, we have $r_{\ell^*}(u_i) \ne r_{\ell^*}(v_i)$ at level ℓ^* in the corresponding tree. Thus each such pair will contribute 2 to $\left\|\mathrm{PA}_{\ell^*}^d(x') - \mathrm{PA}_{\ell^*}^d(y')\right\|_1$. The issue is that it is possible that for those pairs (u_i, v_i) with $\|u_i - v_i\|_2 < A$, we have $r_{\ell^*}(u_i) \ne r_{\ell^*}(v_i)$, and each such pair will also contribute 2 to $\left\|\mathrm{PA}_{\ell^*}^d(x') - \mathrm{PA}_{\ell^*}^d(y')\right\|_1$. In this case we say such a pair (u_i, v_i) is misclassified. If this happens then our algorithm may try to "recover" a pair whose distance is not in the top-k. This in itself is not a problem since recovering such pairs will only decrease the resulting EMD. The problem is that although Bob knows $r_{\ell^*}(u_i)$, he still do not know the exact location of u_i to where he should relocate v_i. The current algorithm simply relocates v_i to an arbitrary leaf in the subtree rooted at $r_{\ell^*}(u_i)$, but doing that will introduce an error of at most A, which could be larger than $\|u_i - v_i\|_2$ itself. To handle this we introduced a random shift with a distance δ in our algorithm. The hope is that such mis-classifications will not happen too often.

First, we need the following observation. The proof can be found in B.1.

Proposition 1. *In dimension d, for a line with arbitrary but fixed slope and length x, the probability of the line cut by a grid of side length K is at most $\frac{\sqrt{d}x}{K}$.*

Set $\eta = \frac{\alpha-\beta}{4\sqrt{d}}$. We focus on a level ℓ such that $2^{L-\ell} = \sum_{i=\beta k}^{n} \|u_i - v_i\|_2 / (\eta k)$ [8], and will show that with probability $3/4$, $\left\|\mathrm{PA}_\ell^d(x') - \mathrm{PA}_\ell^d(y')\right\|_1 \le 2\alpha k$. If this is the case, then according to the definition of ℓ^*, it holds that $A = \sqrt{d} \cdot 2^{L-\ell^*} \le \sqrt{d} \cdot 2^{L-\ell}$.

For $i = \beta k, \ldots, n$, let T_i be the indicator variable of the event that (u_i, v_i) is misclassified at level ℓ. Then by Proposition 1, we have that $\Pr[T_i = 1] \le \frac{\sqrt{d}\|u_i - v_i\|_2}{2^{L-\ell}}$. Let

[7] We are not trying to optimize constants here.

[8] For convenience, we assume that $\sum_{i=\beta k}^{n} \|u_i - v_i\|_2 / (\eta k)$ is a power of 2. Such an assumption will not change the approximation ratio by more than a factor of 2.

$T = \sum_{\beta k}^{n} T_i$. We have

$$\mathbb{E}[T] \leq \sum_{i=\beta k}^{n} \frac{\sqrt{d}\,\|u_i - v_i\|_2}{2^{L-\ell}} = \frac{\sum_{i=\beta k}^{n} \sqrt{d}\,\|u_i - v_i\|_2}{\sum_{i=\beta k}^{n} \|u_i - v_i\|_2 / (\eta k)} = \sqrt{d}\eta k.$$

By Markov inequality, we have $T \leq 4\sqrt{d}\eta k = (\alpha - \beta)k$ with probability $3/4$. Therefore with probability $3/4$, it holds that

$$\left\| \mathrm{PA}_\ell^d(x') - \mathrm{PA}_\ell^d(y') \right\|_1 \leq 2(T + \beta k) \leq 2\alpha k.$$

Consequently, with probability $3/4$,

$$A \leq \sqrt{d} \cdot \sum_{i=\beta k}^{n} \|u_i - v_i\|_2 / (\eta k) = \frac{4d}{(\alpha - \beta)k} \sum_{i=\beta k}^{n} \|u_i - v_i\|_2 . \tag{1}$$

Now suppose that (1) holds. The optimal algorithm (OPT) will correct the first k pairs, leaving the rest pairs untouched. Thus

$$\mathrm{EMD}(x, y_{\mathtt{OPT}}) \geq \sum_{i=k+1}^{n} \|u_i - v_i\|_2 \geq \sum_{i=\beta k}^{n} \|u_i - v_i\|_2 .$$

In our algorithm (SOL), the first αk pairs are recovered so that the distance between each such pair is at most A after the relocation. Let n_1 be the largest number such that $\|u_{n_1} - v_{n_1}\|_2 \geq A$. The first n_1 pairs always get recovered since the original distance between each such pair is at least A. Therefore,

$$\begin{aligned} \mathrm{EMD}(x, y^*) &\leq \alpha k \cdot A + \sum_{i=n_1+1}^{n} \|u_i - v_i\|_2 \\ &\leq \alpha k \cdot A + \max\{\beta k - n_1, 0\} \cdot A + \sum_{i=\beta k}^{n} \|u_i - v_i\|_2 . \end{aligned}$$

Thus

$$\begin{aligned} \frac{\mathrm{EMD}(x, y^*)}{\mathrm{EMD}(x, y_{\mathtt{OPT}})} &\leq \frac{\alpha k \cdot A + \max\{\beta k - n_1, 0\} \cdot A + \sum_{i=\beta k}^{n} \|u_i - v_i\|_2}{\sum_{i=\beta k}^{n} \|u_i - v_i\|_2} \\ &\leq 1 + \frac{(\alpha + \beta)Ak}{\sum_{i=\beta k}^{n} \|u_i - v_i\|_2} \\ &\leq 1 + \frac{4d(\alpha + \beta)}{\alpha - \beta} \qquad \text{(by (1))} \\ &= 1 + 8d < C. \end{aligned}$$

Therefore with probability $(3/4 - \epsilon \cdot \log(2\varDelta)) \geq 2/3$ our algorithm achieves a C-approximation. The first term $3/4$ is the probability that Equation 1 holds, and the second error term is introduced by applying Lemma 1 (choose $\epsilon = 1/(12\log(2\varDelta))$) to each of the $\log(2\varDelta)$ levels of the pyramid arrays.

Communication complexity. We need $\log(2\Delta)$ encodings from Lemma 1 with $N = \Delta^d$, $t = 2\alpha k$ and $\epsilon = 1/(12\log(2\Delta))$. Each such encoding has a length of $O(\log(1/\epsilon)\log(nN) + t\log(nN)) = O((k + \log\log\Delta)\log(n\Delta^d))$ bits. We also need an additional $O(d\log\Delta)$ bits of communication to transmit the δ, the distance of the random shift. Thus the total cost is $O((k+\log\log\Delta)\log\Delta\log(n\Delta^d)) = \tilde{O}(k\log\Delta\log(n\Delta^d))$.

3 The Lower Bound

In this section, we show that any randomized communication protocol that computes a C-approximation for d-dimensional EMD k-Budget Error-Correcting has communication complexity $\Omega(k\frac{\log\Delta}{\log C}(d\log\Delta - \log k)$. The proof is a reduction from the two-party one-way communication problem Augmented Indexing.

Definition 3 (Augmented Indexing). *Let $X = (X_1, \ldots, X_n)$ where $X \in \mathcal{U}^n$ for some universe $\mathcal{U}$. Let $I \in [n]$. Alice is given X, Bob is given I and $(X_{I+1}, \ldots, X_n)$. Alice sends message M_{AI} to Bob and upon reception Bob outputs X_I.*

Intuitively, since Alice does not know the index I, it is impossible to solve Augmented Indexing with a message of size $o(|X|)$. In [17] it is shown that the uniform distribution on X is a hard distribution for a version of Augmented Indexing where Bob also holds some $Y \in \mathcal{U}$ and the goal is to output 1 if $X_I = Y$ and 0 otherwise. They show that $\Omega(n\log|\mathcal{U}|)$ communication is necessary for protocols with error at most $\frac{1}{4|\mathcal{U}|}$. In our version of Augmented Indexing , Bob has to learn X_I. This allows us to modify (actually simplify) the proof in [17] to obtain the same communication bound for constant error. The proof of the following lemma can be found in Appendix B.2.

Lemma 2. *If X and I are chosen uniformly at random and the failure probability of the protocol is at most $1/3$, then $\mathbb{E}_X|M_{\mathrm{AI}}| = \Omega(n\log|\mathcal{U}|)$.*

In the following, we will show how to solve Augmented Indexing with a protocol for d-dimensional EMD k-Budget Error-Correcting. In our application, the universe $\mathcal{U}$ from which the elements of X are chosen is a large family of k-subsets of a set $[p]$ $(p > k)$ with bounded intersection. For $\epsilon > 0$, we define $\mathcal{C}_{k,p}^{\epsilon k}$ to be a family of k-subsets of $[p]$ such that any two subsets have at most $k(1-\epsilon)$ elements in common. Then we will use $\mathcal{U} = \mathcal{C}_{k,p}^{k/100}$. Such a family is equivalent to a binary constant weight code of length p, weight k, and distance $2\epsilon k$. In Lemma 3 we show that there is a large set of k-subsets with bounded intersection. The proof can be found in Appendix B.3.

Lemma 3. *Let k, p be integers such that $k < p/2$, and let $\epsilon < 1 - 1/(\lfloor p/k \rfloor)$. Then there is a family $\mathcal{C}_{k,p}^{\epsilon k}$ of k-subsets of $[p]$ such that for $c_1, c_2 \in \mathcal{C}_{k,p}^{\epsilon k}, c_1 \neq c_2 : |c_1 \cap c_2| \leq k(1-\epsilon)$ and*

$$|\mathcal{C}_{k,p}^{\epsilon k}| \geq (\lfloor p/k \rfloor)^{k(1-H_{\lfloor p/k \rfloor}(\epsilon))},$$

where H_q is the q-ary entropy function $H_q(x) = -x\log_q\frac{x}{q-1} - (1-x)\log_q(1-x)$.

Suppose that the EMD k-Budget Error-Correcting protocol outputs a C-approximation. We take three integer parameters L, p, k such that $p > k$ and $L = \lceil\frac{\log(p^{1/d}/10)}{\log(200C)+2}\rceil$. Let

$X = (X_1, \ldots, X_L)$ where $X_i \in \mathcal{U} = \mathcal{C}_{k,p}^{k/100}$. X_i is a k-subset of $[p]$ and we write $X_i = (X_i^1, \ldots, X_i^k)$.

Consider the Augmented Indexing problem where Alice has X, Bob has $I \in [L]$ and $(X_{I+1}, \ldots, X_L)$. Then by applying Lemma 2 and Lemma 3, the communication complexity of this problem is

$$\begin{aligned}\Omega(L \cdot \log |\mathcal{U}|) &= \Omega\left(L \cdot \log(|\mathcal{C}_{k,p}^{k/100}|)\right) \\ &= \Omega\left(\frac{\log(p^{1/d})}{\log C} \cdot \log\left((\lfloor p/k \rfloor)^{k\left(1-H_{\lfloor p/k \rfloor}(1/100)\right)}\right)\right) \\ &= \Omega\left(\frac{k}{d}\frac{\log p}{\log C} \log\left(\frac{p}{k}\right)\right), \qquad (2)\end{aligned}$$

where we used the fact that for any $q \geq 1, H_q(1/100) < 0.35$.

Reduction. Given such an Augmented Indexing instance, Alice and Bob construct a d-dimensional instance for EMD k-Budget Error-Correcting with grid $[\Delta]^d$ $(\Delta = p^{2/d})$ and $n = 10kpL$ points. The construction requires a parameter B which we set to be $\log(200C) + 2$. Furthermore, we make use of the set of coordinates $Z_p = \{1, p^{1/d} + 1, 2p^{1/d} + 1, \ldots (p^{1/d} - 1)p^{1/d} + 1\}^d$ that we call *point centers* since in the reduction Alice and Bob place many points onto these coordinates. Note that $|Z_p| = p$.

The reduction consists of 3 steps.

Step 1. Alice and Bob use an arbitrary but fixed bijection $f : [p] \to Z_p$. They proceed as follows to set up the EMD k-Budget Error-Correcting instance:

1. Alice and Bob place $\frac{n}{p}$ points to each point center in Z_p.
2. For each X_i^j $(i \in [L], j \in [k])$, Alice moves one point from point center $f(X_i^j)$ by a distance of 2^{Bi}, resulting a new point at location $f(X_i^j) + 2^{Bi}e_1$, where e_1 is the d-dimensional standard basis unit vector pointing to dimension 1. Bob does the same for each X_i^j with $i > I$. Denote Alice's points set by x and Bob's points set by y. Since $n = 10kpL$, there will be $n/p = 10kL$ points on each point center. Thus Alice and Bob can ensure that there are always enough points to move.

Here, the effect of parameter B becomes clear: Alice and Bob displace points from the point centers by distances 2^{Bi}. B is hence responsible for increasing the distance of points that correspond to different values of i. Note that we set $B = \Theta(\log C)$, hence the distances increase as the approximation factor increases.

Step 2. Alice and Bob run the protocol for EMD k-Budget Error-Correcting. Let y^* denote the points of Bob after the relocation outputted by the protocol.

Step 3. Bob rounds the points y^* to the closest positions in $\{Z_p + 2^{Bi}e_1 \mid i \in [L]\}$. Then, he computes an estimate $\tilde{X}'_I$ of X_I as follows: if there is a point in y^* at position $x + 2^{BI}e_1$ for some $x \in Z_p$, then $f^{-1}(x) \in \tilde{X}'_I$. Next, Bob selects $\tilde{X}_I \in \mathcal{C}_{k,p}^{k/100}$ such that $|\tilde{X}_I \cap \tilde{X}'_I|$ is maximized.

Theorem 2. *Any randomized communication protocol that computes a C-approximation to EMD k-Budget Error-Correcting on d-dimensional grid $[\Delta]^d$ with probability $2/3$ requires a message of size $\Omega(k\frac{\log\Delta}{\log C}(d\log\Delta - \log k))$, assuming that $k < \Delta^{d/2}$.*

Proof. We use the prior reduction from Augmented Indexing to EMD k-Budget Error-Correcting. Recall that the setup for the EMD k-Budget Error-Correcting instance uses $p = \Delta^{d/2}$, $B = \log(200C) + 2$, $L = \lceil\log(p^{1/d}/10)/B\rceil$, and $n = 10kpL$.

Firstly, note that the distance between any two point centers is larger than or equal to $\sqrt{\Delta}$. The maximal distance that a point is displaced from its point center is $2^{BL} = 0.1p^{1/d} = 0.1\sqrt{\Delta}$. Under this condition, the EMD between Alice's and Bob's points is the sum of the distances of the points that only Alice moved, that is, $\mathrm{EMD}(x, y) = k\sum_{i=1}^{I} 2^{Bi}$. Furthermore, we have $\min_{\tilde{y}\in\mathcal{N}_k(y)} \mathrm{EMD}(x, \tilde{y}) = k\sum_{i=1}^{I-1} 2^{Bi}$, which can be obtained by correcting the k points that Alice moved by a distance of 2^{BI}. Since our protocol approximates the EMD within a factor C, we obtain $\mathrm{EMD}(x, y^*) \leq C \cdot k\sum_{i=1}^{I-1} 2^{Bi}$. Let $\mathrm{err} = |X_I \setminus \tilde{X}_I|$ be the number of points that Bob failed to recover. Then each of these points contributes to the EMD by at least $(2^{BI} - 2^{B(I-1)})/2$, since these points got rounded to some index other than I. We obtain

$$\mathrm{err} \cdot \left(2^{BI} - 2^{B(I-1)}\right)\big/\, 2 \leq C \cdot k\sum_{i=1}^{I-1} 2^{Bi},$$

Therefore we conclude that $\mathrm{err} < \frac{Ck}{2^{B-2}} = \frac{k}{200}$. Since the k-subsets of $\mathcal{C}_{k,p}^{k/100}$ differ by at least $\frac{k}{100}$ elements, we can recover $\tilde{X}_I^i = X_I$. The lower bound for EMD k-Budget Error-Correcting follows by plugging $p = \Delta^{d/2}$ into Equation (2).

References

1. A. Andoni, K. Do Ba, P. Indyk, and D. Woodruff. Efficient sketches for earth-mover distance, with applications. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 324–330. IEEE, 2009.
2. Z. Bar-Yossef. *The complexity of massive data set computations*. PhD thesis, University of California at Berkeley, 2002.
3. E. R. Berlekamp. *Algebraic coding theory*, volume 111. McGraw-Hill New York, 1968.
4. C. Chefd'hotel and G. Bousquet. Intensity-based image registration using earth mover's distance. In *SPIE*, 2007.
5. K. L. Clarkson and D. P. Woodruff. Numerical linear algebra in the streaming model. In *STOC '09*, pages 205–214. ACM, 2009.
6. G. Cormode, M. Paterson, S. C. Sahinalp, and U. Vishkin. Communication complexity of document exchange. In *SODA '00*, pages 197–206. SIAM, 2000.
7. K. Do Ba, P. Indyk, E. Price, and D. P. Woodruff. Lower bounds for sparse recovery. In *SODA '10*, pages 1190–1197. SIAM, 2010.
8. J. Feigenbaum, S. Kannan, M. J. Strauss, and M. Viswanathan. An approximate l 1-difference algorithm for massive data streams. *SIAM Journal on Computing*, 32(1):131–151, 2002.
9. A. Gilbert and P. Indyk. Sparse recovery using sparse matrices. *Proceedings of the IEEE*, 98(6):937–947, 2010.
10. K. Grauman and T. Darrell. Fast contour matching using approximate earth movers distance. In *CVPR*, pages 220–227, 2004.

11. A. S. Holmes, C. J. Rose, and C. J. Taylor. Transforming pixel signatures into an improved metric space. *Image Vision Comput.*, 20(9-10):701–707, 2002.
12. W. C. Huffman and V. Pless. *Fundamentals of error-correcting codes*. Cambridge university press, 2003.
13. P. Indyk. A near linear time constant factor approximation for euclidean bichromatic matching (cost). In *SODA '07*, pages 39–42. SIAM, 2007.
14. P. Indyk and N.Thaper. Fast color image retrieval via embeddings. *Workshop on Statistical and Computational Theories of Vision (at ICCV)*, 2003.
15. P. Indyk and E. Price. K-median clustering, model-based compressive sensing, and sparse recovery for earth mover distance. In *STOC*, pages 627–636, 2011.
16. U. Irmak, S. Mihaylov, and T. 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.
17. T. S. Jayram and D. Woodruff. Optimal bounds for johnson-lindenstrauss transforms and streaming problems with sub-constant error. In *SODA '11*, pages 1–10. SIAM, 2011.
18. H. Jowhari. Efficient communication protocols for deciding edit distance. In *ESA*, 2012.
19. D. M. Kane, J. Nelson, and D. P. Woodruff. On the exact space complexity of sketching and streaming small norms. In *SODA '10*, pages 1161–1178. SIAM, 2010.
20. J. Massey. Shift-register synthesis and bch decoding. *Information Theory, IEEE Transactions on*, 15(1):122–127, 1969.
21. J. Puzicha, J. M. Buhmann, Y. Rubner, and C. Tomasi. Empirical evaluation of dissimilarity measures for color and texture. In *ICCV*, pages 1165–1173, 1999.
22. Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision*, 40(2):99–121, Nov. 2000.
23. E. Verbin and Q. Zhang. Rademacher-sketch: A dimensionality-reducing embedding for sum-product norms, with an application to earth-mover distance. In *ICALP (1)*, pages 834–845, 2012.
24. Y. Yoo and T. Park. A moving object detection algorithm for smart cameras. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

A Proof for Lemma 1

Lemma 1. *Let x and y be two arrays of length N each. Each element in x and y is from $\{0, \ldots, n\}$. x could be encoded in $O(\log(1/\epsilon)\log(nN) + t\log(nN))$ bits, with which one having y in hand can*

- *fully decode x when $\|x - y\|_1 \leq t$; or*
- *output "impossible" when $\|x - y\|_1 > t$.*

with probability $1-\epsilon$ in encoding time $O(N^2 \log^2 n)$ and decoding time $O(N^2 \log^2 n)$.

Proof. First, the encoding of x contains the ℓ_1 sketch of x in size $O(\log(1/\epsilon)\log(nN))$ bits by using the following lemma from [8].

Lemma 4 **(ℓ_1 Sketch [8]).** *Let x and y be two vectors of length N containing elements from $\{0, \ldots, n\}$. By having a sketch of x with $O(\log(Nn)\log(1/\epsilon)/\delta^2)$ bits, one can compute a distance in the interval $[(1-\delta)\|x-y\|_1, (1+\delta)\|x-y\|_1]$ with probability $1-\epsilon$.*

Using this sketch with $\delta = 1/2$, one having y in hand can know $\|x - y\|_1$ up to a factor of $1 \pm \frac{1}{2}$. Let the distance computed be X. If $X > \frac{3}{2}t$ then we already know that the output should be "impossible" since $\|x - y\|_1 > t$ with probability $(1 - \epsilon)$ in this case. Otherwise we know $\frac{3}{2}t \geq X \geq \frac{1}{2}\|x - y\|_1$ implying $\|x - y\|_1 \leq 3t$ with probability $(1 - \epsilon)$.

Let $\bar{x}$ and $\bar{y}$ be the binary representation of x and y, respectively. That is, $\bar{x} = \{\bar{x}_1, \ldots, \bar{x}_N\}$ where $\bar{x}_i$ is the $\log(n + 1)$-bit binary representation of x_i, and similar for $\bar{y}$. The encoding also contains the error-correcting part for $\bar{x}$ in the Reed-Solomon code in the following lemma, denoted by c, with $r = O(\log(N \log(n + 1)))$, $d = 6t \cdot \left(\left\lfloor \frac{\log(n+1)}{\log(N \log(n+1))} \right\rfloor + 1\right) + 1$ and $l = \frac{N \log(n+1)}{r}$.

Lemma 5 (Reed-Solomon Code [3, 20]). *For a message $\mathbb{A}$ of l elements from $GF(2^r)$, there is a systematic error-correcting code of $l + 2d + 1 < 2^r$ elements which can be used to recover $\mathbb{A}$ if there are $\leq d$ lost elements in the coding. Moreover, the code is of the form: ($\mathbb{A}$, $\mathbb{G}$), where $\mathbb{G}$ is the "error-correcting" part consisting of $2d$ elements. The encoding and decoding time of the code is $O((l + d)^2)$.*

The length of c is

$$\left(6t \cdot \left(\left\lfloor \frac{\log(n + 1)}{r} \right\rfloor + 1\right) + 1\right) \cdot r = O(t \log(nN)).$$

Now one tries to recover $\bar{x}$ (also x) using $\bar{y}$ and c. Since $\|x - y\|_1 \leq 3t$, and each different coordinate between x and y will only expand to a block of $\log(n+1)$ consecutive bits in $\bar{x}$, thus the total number of errors in $\bar{x}$ is at most $\lfloor \frac{\log(n+1)}{r} \rfloor + 1$. Therefore, by using c, one can correct up to $3t \cdot \left(\left\lfloor \frac{\log(n+1)}{r} \right\rfloor + 1\right)$ errors in $\bar{x}$, which also means that one can correct up to $3t$ errors in x with c. By taking y as x with $\leq 3t$ errors, we know that Bob can decode x by using c.

The total length of the encoding is $O(\log(1/\epsilon) \log(nN) + t \log(nN))$. And the encoding/decoding process costs $O(N^2 \log^2 n)$. □

B Other Omitted Proofs

B.1 Proof for Proposition 1

Proof. Let the projection length of x on the dimensions be $x_1, x_2, \ldots, x_d$ where $x_1^2 + \cdots + x_d^2 = x^2$. Thus the probability is no more than

$$\sum_{i=1}^{d} \frac{x_i}{A} \leq \frac{1}{A} \sqrt{d \sum_{i=1}^{d} x_i^2} = \frac{\sqrt{d}x}{A}.$$

B.2 Proof for Lemma 2

Proof. The proof follows [17], and uses the standard tools from information complexity. We refer readers to [2] for an introduction of information complexity. Let

$X = (X_1, \ldots, X_n)$ where X_i is chosen uniformly and independently of $(X_j)_{j \neq i}$ from $\mathcal{U}$. Since $\mathbb{E}_X|M_{\text{AI}}| \geq \mathrm{H}(M_{\text{AI}}) \geq \mathrm{I}(X : M_{\text{AI}})$, it is enough to bound $\mathrm{I}(X : M_{\text{AI}})$. Then, by the chain rule for mutual information, and the definition of mutual information we obtain

$$\begin{aligned}\mathrm{I}(X : M_{\text{AI}}) &= \sum_{i=1}^{n} \mathrm{I}(X_i : M_{\text{AI}} \,|\, X_{i+1}, \ldots, X_n) \\ &= \sum_{i=1}^{n} \mathrm{H}(X_i \,|\, X_{i+1}, \ldots, X_n) - \sum_{i=1}^{n} \mathrm{H}(X_i \,|\, M_{\text{AI}}, X_{i+1}, \ldots, X_n).\end{aligned}$$

By independence, we can simplify for all $i \in \{1, \ldots, n\}$ as follows

$$\mathrm{H}(X_i \,|\, X_{i+1}, \ldots, X_n) = \mathrm{H}(X_i) = \log(|\mathcal{U}|).$$

It remains to upper bound $\mathrm{H}(X_i \,|\, M_{\text{AI}}, X_{i+1}, \ldots, X_n)$ for all $i \in \{1, \ldots, n\}$. Note that $\{M_{\text{AI}}, X_{i+1}, \ldots, X_n\}$ is exactly Bob's input for Augmented Indexing. Bob outputs X_i with error ϵ, hence $M_{\text{AI}}, X_{i+1}, \ldots, X_n$ is a predictor for X_i with error probability ϵ. We apply Fano's Inequality and obtain

$$\mathrm{H}(X_i \,|\, M_{\text{AI}}, X_{i+1}, \ldots, X_n) \leq \mathrm{H}(\epsilon) + \epsilon \cdot \log(|\mathcal{U}| - 1),$$

where $\mathrm{H}(\epsilon)$ denotes the binary entropy of ϵ. Combining and setting $\epsilon = 1/3$, we obtain $\mathrm{I}(X : M_{\text{AI}}) = \Omega(n \log |\mathcal{U}|)$.

B.3 Proof for Lemma 3

Lemma 6 (Gilbert-Varshamov Bound [12]). *Let $A_q(M, d)$ be the maximum possible size of a q-ary code with length M and Hamming distance at least d. Then,*

$$A_q(M, d) \geq \frac{q^M}{\sum_{j=0}^{d-1} \binom{M}{j} (q-1)^j}.$$

Proof. We follow the proof of Lemma 3.1 of [7]. Let T be a code of block length k, alphabet $\{1, \ldots, \lfloor p/k \rfloor\}$ and Hamming distance ϵk. From T we obtain a binary code T' with block length p and Hamming distance $2\epsilon k$ by replacing each character i with the $\lfloor p/k \rfloor$-long standard basis vector e_i. Note that T' has exactly k ones. The set $\mathcal{C}_{k,p}^{\epsilon k}$ is obtained by interpreting the code words of T' as the characteristic vectors of the subsets. Then if code words $t_1', t_2' \in T'$ have Hamming distance $2\epsilon k$ then the corresponding k-subsets c_1, c_2 obtained from t_1', t_2' are such that $|c_1 \cap c_2| = k(1 - \epsilon)$. By the Gilbert-Varshamov bound (Lemma 6) we obtain

$$|\mathcal{C}_{k,p}^{\epsilon k}| = |T| \geq \frac{(\lfloor p/k \rfloor)^k}{\sum_{i=0}^{\epsilon k - 1} \binom{k}{i} (\lfloor p/k \rfloor - 1)^i}.$$

Following [7], for $\epsilon < 1 - 1/(\lfloor p/k \rfloor)$ we can use $\sum_{i=0}^{\epsilon k-1} \binom{k}{i} (\lfloor p/k \rfloor - 1)^i < (\lfloor p/k \rfloor)^{H_{\lfloor p/k \rfloor}(\epsilon) k}$, and the result follows.