Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 69 (2022)

List-Decoding XOR Codes Near the Johnson Bound

Silas Richelson* Sourya Roy†

Abstract

In a breakthrough result, Ta-Shma described an explicit construction of an almost optimal binary code (STOC 2017). Ta-Shma's code has distance $\frac{1-\varepsilon}{2}$ and rate $\Omega\big(\varepsilon^{2+o(1)}\big)$ and thus it almost achieves the Gilbert-Varshamov bound, except for the $o(1)$ term in the exponent. The prior best list-decoding algorithm for (a variant of) Ta-Shma's code achieves is due to Alev et al (STOC 2021). This algorithm makes use of SDP hierarchies, and is able to recover from a $\frac{1-\rho}{2}-$fraction of errors as long as $\rho \geq 2^{\log(1/\varepsilon)^{1/6}}$. In this work we give an improved analysis of a similar list-decoding algorithm. Our algorithm works for Ta-Shma's original code, and it is able to list-decode almost all the way to the Johnson bound: it can recover from a $\frac{1-\rho}{2}-$fraction of errors as long as $\rho \geq 2\sqrt{\varepsilon}$.

1 Introduction

Error correcting codes (ECCs) allow a sender to encode a message so that the receiver can recover the full message even if some codeword bits are lost or flipped during transmission. A binary (linear) code is a (linear) map $\mathcal{C} : \{0,1\}^k \to \{0,1\}^n$ which sends $m \in \{0,1\}^k$ to the codeword $\mathcal{C}(m) \in \{0,1\}^n$. Two important parameters of a code are the *distance* and *rate*, which are respectively measures of the code's quality and efficiency. *Rate* is the ratio k/n, the number of message bits per codeword bit; *distance* is the minimum fraction of coordinates on which two distinct codewords disagree. Explicit constructions of asymptotically good codes (codes with constant distance and constant rate) have been known since the 70s [Jus72]. One of the holy grails of modern coding theory is to construct a code with the optimal tradeoff between distance and rate. It is known that codes with optimal distance $\varepsilon = 1/2$ must have exponentially small rate [Plo60]. The Gilbert-Varshamov (GV) bound [Gil52, Var57] states for any $\varepsilon \in \big(0, 1/2\big)$, there exists a code with distance ε and rate $1 - H(\varepsilon) - o_n(1)$ where $H(\cdot)$ is Shannon's binary entropy function. This is proved non-constructively, we do not know of an explicit construction which achieves the GV bound. For distances ε close to $1/2$, the GV bound says that there exists a code with distance $\frac{1-\varepsilon}{2}$ and rate $\Omega(\varepsilon^2)$. On the other hand, it is known that any code with distance $\frac{1-\varepsilon}{2}$ must have rate $\mathcal{O}\big(\varepsilon^2 \cdot \log(1/\varepsilon)\big)$ [ABN+92].

Most applications of ECCs require an efficient decoding algorithm which recovers the message $m \in \{0,1\}^k$ from $y \in \{0,1\}^n$, provided the Hamming distance, $\Delta\big(y, \mathcal{C}(m)\big) := \Pr_{i\sim[n]}\big[y_i \neq \mathcal{C}(m)_i\big]$ is at most $\frac{1-\rho}{2}$ for some parameter $\rho > 0$. In the worst-case error model, unique decoding becomes impossible as soon as the decoding radius is at least half of the distance of the code. In order to handle smaller values of ρ where there might be more than one valid codeword within the Hamming ball of

*University of California, Riverside. Email: silas@cs.ucr.edu. This work was partly done while visiting Bocconi University, supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101019547).

†University of California, Riverside. Email: sroy004@ucr.edu

ISSN 1433-8092

radius $\frac{1-\rho}{2}$, we require the decoding algorithm to output the list of all valid codewords which are close to y. Formally, the *list decoding* problem for an ECC is the following: given $y \in \{0,1\}^n$ and $\rho > 0$, efficiently recover

$$\mathsf{LIST}(y,\rho) := \left\{ m \in \{0,1\}^k : \Delta\big(y, \mathcal{C}(m)\big) \leq \frac{1-\rho}{2} \right\}.$$

The list decoding problem makes sense as long as the set $\mathsf{LIST}(y,\rho)$ is guaranteed not to be too big; certainly if $\big|\mathsf{LIST}(y,\rho)\big|$ is super-polynomial (in n) then it cannot be efficiently recovered. If an ECC has distance $\frac{1-\varepsilon}{2}$ then $\big|\mathsf{LIST}(y,\rho)\big|$ is exponentially large whenever $\rho < \varepsilon$, so the best one can hope for is a list-decoding algorithm which works for $\rho \geq \varepsilon$. It is possible to show, non-constructively, that binary linear codes exist which achieve the GV bound, and for which $\mathsf{LIST}(y,\rho)$ has polynomial size for all $y \in \{0,1\}^n$ whenever $\rho \geq \varepsilon$. Such codes are said to *achieve list decoding capacity*; giving an explicit construction of such a code along with an efficient decoding algorithm is a fantastic open problem.[1] It is not true for general codes of distance $\frac{1-\varepsilon}{2}$ that $\big|\mathsf{LIST}(y,\rho)\big|$ will always be at most polynomial size whenever $\rho \geq \varepsilon$. The best bound we have that holds in general is the *Johnson bound* which ensures that $\big|\mathsf{LIST}(y,\rho)\big| = \mathsf{poly}(k)$ whenever $\rho \geq \sqrt{\varepsilon}$.

Recent Progress. As mentioned above, currently we do not know of an explicit binary linear code which matches the GV bound; constructing such a code has been a major open problem for nearly 70 years. A few years ago, in a major breakthrough, Ta-Shma [Ta-17] gave a construction which got very close: his code achieved rate $\Omega(\varepsilon^{2+o_\varepsilon(1)})$ and distance $\frac{1-\varepsilon}{2}$. Ta-Shma's original paper described only the encoding algorithm, it did not give an efficient decoding algorithm. A decoding algorithm based on semidefinite programming hierarchies has been developed in subsequent work [AJQ+20, JQST20, JST21]. These algorithms can efficiently recover $\mathsf{LIST}(y,\rho)$ as long as $\rho \geq 2^{-\Theta(\log(1/\varepsilon)^{1/6})}$. As $2^{-\Theta(\log(1/\varepsilon)^{1/6})} \gg \sqrt{\varepsilon} > \varepsilon$, there is significant room for improvement.

1.1 Our Main Result

In this work we prove the following theorem.

Theorem 1 (Main). *For any $\varepsilon > 0$, there is an explicit construction of a binary linear code with distance $\frac{1-\varepsilon}{2}$, rate $\Omega\big(\varepsilon^{2+o(1)}\big)$, and which is efficiently list-decodable from a $\frac{1-\rho}{2}$–fraction of errors whenever $\rho \geq 2\sqrt{\varepsilon}$.*

Our theorem is proved by improving a key step in the analysis of a decoding algorithm which has been developed in an extensive (perhaps daunting) tower of prior work. In the next section, we present an overview of the decoding algorithms from prior work in a simplified setting.

1.2 Background – SDP Decoding of Random Walk XOR Codes

Basic Setup. Let A be (the vertex set of) a regular graph, and let RW_A^t denote the set of $t-$length walks in A. So $\mathsf{RW}_A^t = \big\{(a_1,\ldots,a_t) \in A^t : a_i \sim a_{i+1}\ \forall\, i = 1,\ldots,t-1\big\}$, where $a_i \sim a_{i+1}$ indicates that a_i and a_{i+1} are neighbors in A. The *random walk XOR code* sends a message $\mathbf{x} \in \{0,1\}^A$ to the codeword $\mathbf{y} \in \{0,1\}^{\mathsf{RW}_A^t}$ where for $\vec{a} = (a_1,\ldots,a_t) \in \mathsf{RW}_A^t$, $\mathbf{y}_{\vec{a}} = \mathbf{x}_{a_1} \oplus \cdots \oplus \mathbf{x}_{a_t}$, where $\oplus$ denotes

[1] Explicit constructions of codes with large alphabets are known which achieve list-decoding capacity (see [GR08] and the references therein).

XOR. The list decoding problem for this code is: given $\tilde{\mathbf{y}} \in \{0,1\}^{\mathsf{RW}_A^t}$ and $\rho > 0$, algorithmically recover

$$\mathsf{LIST}(\tilde{\mathbf{y}}, \rho) := \left\{ \mathbf{x} \in \{0,1\}^A : \Pr_{\vec{a} \sim \mathsf{RW}_A^t} \left[\tilde{\mathbf{y}}_{\vec{a}} = \mathbf{x}_{a_1} \oplus \cdots \oplus \mathbf{x}_{a_t} \right] > \frac{1+\rho}{2} \right\}.$$

Currently the best decoding algorithms we have are based on semidefinite programming hierarchies. The basic idea is to, given $\tilde{\mathbf{y}} \in \{0,1\}^{\mathsf{RW}_A^t}$, set up and solve a semidefinite program to obtain what has become called a *pseudodistribution* on $\{0,1\}^A$. Our algorithm will use the same SDP as was used in [JST21], details can be found in Section 5. For now, let us jump ahead in the decoding algorithm to the point after the SDP has been solved.

Pseudodistributions. A pseudodistribution can be thought of as an oracle $\mathcal{O}$ which is interacted with through the following interface: we send $\mathcal{O}$ any subset $S \subset A$ of size at most $|S| \leq r$ and $\mathcal{O}$ probabilistically generates and returns a string $\sigma \in \{0,1\}^{|S|}$, which we think of as being an assignment to the elements of S. The properties of the SDP will ensure certain consistency requirements. For example, for any $S \subset A$ of size $|S| \leq r$ and any $a \in S$, the marginal distribution on σ_a is identical to the distribution $\mathcal{O}(a)$ obtained by sending $\{a\}$ to $\mathcal{O}$. The ideal situation would be that $\mathcal{O}$ is returning the size $\leq r$ marginals of some actual distribution on $\{0,1\}^A$. This will be the case when $r = |A|$ but will not be the case in general for smaller r. The runtime of the SDP solver is roughly $|A|^r$, so efficient decoding necessitates choosing $r \ll |A|$, and so $\mathcal{O}$ might not correspond to an actual distribution on $\{0,1\}^A$. For this reason, the next step of the decoding algorithm involves converting $\mathcal{O}$ into an actual distribution on $\{0,1\}^A$ in such a way so that some key properties of the pseudodistribution are maintained. This is called *rounding*, and we discuss it next.

Rounding. Probably the most straightforward way to recover a distribution on $\{0,1\}^A$ from a pseudodistribution is to draw $\sigma_a \sim \mathcal{O}(a)$ for each $a \in A$, and then output $\sigma = (\sigma_a)_{a \in A} \in \{0,1\}^A$. The size-one marginals of this distribution will be identical to the size-one marginals given by $\mathcal{O}$, but larger marginals could vary significantly since the distribution is ignoring all correlations among the bits. This turns out to be a problem in the analysis, since certain arithmetic properties of the pseudodistribution might fail to hold for the distribution. Rounding refers to a different method to produce a distribution which much more accurately imitates $\mathcal{O}$. For simplicity, at this point we describe the rounding method used in [BRS11, AJQ+20, JQST20, JST21] (ours will be slightly different due to some technicalities which arise in the analysis). The rounding distribution in these prior works works as follows.

- First, a subset $S \subset A$ of size $|S| < r$ and a string $\sigma \sim \mathcal{O}(S)$ are drawn; the S part of the output string is set to σ (the pair (S, σ) is called a *slice*).
- Then for each $a \notin S$, $\tau \sim \mathcal{O}(S \cup \{a\})$ is drawn such that $\tau|_S = \sigma$, and the $a-$th bit of the output string is set to τ_a.

So each bit of the distribution is drawn independently from $\mathcal{O}$, subject to all of the bits being correlated with (S, σ).

Notation. Let $\mathcal{O}^{(S,\sigma)}$ denote the pseudodistribution where for a set $T \subset A$ such that $|T \cup S| \leq r$, the distribution $\mathcal{O}^{(S,\sigma)}(T)$ draws $\tau \sim \mathcal{O}(S \cup T)$ conditioned on $\tau|_S = \sigma$, and outputs $\tau|_T \in \{0,1\}^T$. The consistency conditions mentioned above will imply that for all $S, T \subset A$ such that $|S \cup T| \leq r$ the T marginal of the distribution $\mathcal{O}(S \cup T)$ is identical to the distribution which draws $\sigma \sim \mathcal{O}(S)$ and outputs $\tau \sim \mathcal{O}^{(S,\sigma)}(T)$. As has become the custom when working with pseudodistributions, we call

expectations of random variables which are defined over the "local" distributions $\{\mathcal{O}(S)\}_S$ *pseudoexpectations*, and we denote them $\tilde{\mathbb{E}}$. So, for example, if $\{\mathsf{X}_S\}_S$ is a collection of random variables where each X_S draws $\sigma \sim \mathcal{O}(S)$ and outputs $\mathsf{X}_S(\sigma)$, then we write $\tilde{\mathbb{E}}[\mathsf{X}_S]$ instead of $\mathbb{E}_{\sigma\sim\mathcal{O}(S)}\big[\mathsf{X}_S(\sigma)\big]$. For a slice (S,σ), we denote by $\tilde{\mathbb{E}}^{(S,\sigma)}$ the *conditional pseudoexpectation* corresponding to the conditional pseudodistribution $\mathcal{O}^{(S,\sigma)}$.

The Decoding Algorithm and Analysis Overview. The decoding algorithm works as follows given $\tilde{\mathbf{y}} \in \{0,1\}^{\mathsf{RW}_A^t}$:

1. it sets up and solves an SDP obtaining a pseudodistribution $\mathcal{O}$;
2. it chooses a slice (S,σ) and for each $a \in A$, draws $\alpha \sim \mathcal{O}^{(S,\sigma)}(a)$ and sets $\tilde{\mathbf{x}}_a = \alpha$;
3. it outputs $\tilde{\mathbf{x}} \in \{0,1\}^A$.

The analysis involves showing that for all $\mathbf{x} \in \mathsf{LIST}(\tilde{\mathbf{y}},\rho)$, the decoding algorithm outputs $\tilde{\mathbf{x}}$ which has good agreement with $\mathbf{x}$ with non-negligible probability. Our final construction makes use of an inner "base code" to recover $\mathbf{x}$ from $\tilde{\mathbf{x}}$, and also repeats this procedure to recover $\mathbf{x}$ with high probability, rather than with non-negligible probability. So what we need to show is that for any $\mathbf{x} \in \mathsf{LIST}(\tilde{\mathbf{y}},\rho)$, a non-negligible fraction of the slices (S,σ) are such that

$$\mathbb{E}_{a\sim A}\Big[\tilde{\mathbb{E}}^{(S,\sigma)}\big[(-1)^{\mathbf{x}_a\oplus\alpha}\big]\Big] \geq \rho_{\mathsf{base}}, \tag{4}$$

where $\alpha \sim \mathcal{O}^{(S,\sigma)}(a)$ is implied by the notation $\tilde{\mathbb{E}}^{(S,\sigma)}$, and where ρ_{base} is the list-decoding radius of the base code, of which $\mathbf{x} \in \{0,1\}^A$ is a codeword.[2] This enables recovering $\mathbf{x}$ using the base code's list decoding algorithm. The properties of the SDP will ensure that

$$\mathbb{E}_{\vec{a}\sim\mathsf{RW}_A^t}\Big[\tilde{\mathbb{E}}\big[(-1)^{\alpha_1\oplus\cdots\oplus\alpha_t\oplus\tilde{\mathbf{y}}_{\vec{a}}}\big]\Big] \geq \rho \tag{1}$$

holds, where $\vec{\alpha} = (\alpha_1,\ldots,\alpha_t) \sim \mathcal{O}(\vec{a})$ is implied by the $\tilde{\mathbb{E}}$ notation. This will follow easily from the optimality of the pseudodistribution recovered by the SDP solver and the assumption that some valid codeword exists within distance $\frac{1-\rho}{2}$ of $\tilde{\mathbf{y}}$. In words, (1) says that the random variable which draws $\vec{a} \sim \mathsf{RW}_A^t$ and $\vec{\alpha} \sim \mathcal{O}(\vec{a})$ and outputs $\alpha_1 \oplus \cdots \oplus \alpha_t \in \{0,1\}$ will have good agreement with the corrupted codeword $\tilde{\mathbf{y}} \in \{0,1\}^{\mathsf{RW}_A^t}$. The derivation of (4) from (1) from [AJQ+20] (and subsequent works) proceeds in three steps. First, it is shown that (1) implies

$$\mathbb{E}_{\vec{a}\sim\mathsf{RW}_A^t}\Big[\tilde{\mathbb{E}}\big[(-1)^{(\alpha_1\oplus\mathbf{x}_{a_1})\oplus\cdots\oplus(\alpha_t\oplus\mathbf{x}_{a_t})}\big]\Big] \geq \rho'. \tag{2}$$

Namely, the pseudodistribution has good agreement with the valid codewords which are close to $\tilde{\mathbf{y}}$. This is arranged by having the SDP minimize a certain "convex entropy proxy" which decreases as the pseudodistribution is made to agree with more valid codewords. The next step is to show that (2) implies that with non-negligible probability over the slice (S,σ),

$$\mathbb{E}_{\vec{a}\sim\mathsf{RW}^t}\Big[\tilde{\mathbb{E}}^{(S,\sigma)}\big[(-1)^{(\alpha_1\oplus\mathbf{x}_{a_1})\oplus\cdots\oplus(\alpha_t\oplus\mathbf{x}_{a_t})}\big]\Big] \geq \rho''. \tag{3}$$

[2]The random walk XOR construction amplifies distance so it is critical to assume that $\mathbf{x}$ is already a codeword of some good, but not optimal code.

This step uses a theorem of [BRS11] as a blackbox. In this work we improve this part of the analysis by tailoring the techniques of [AKK+08, BRS11] to the specific setting of decoding random walk XOR codes. We will discuss this in more detail momentarily. The final step involves deriving (4) from (3). This requires proving that random walks on expander graphs are good *parity samplers*. This is proved in [Bog12], and more recently Ta-Shma [Ta-17] proved the same for wide replacement product walks which was a key component of his breakthrough construction of nearly optimal binary codes.

Parameters. The above discussion has introduced parameters $\rho_{\mathsf{base}}, \rho, \rho', \rho''$. The way to think about these parameters is as follows: ρ_{base} is non-negotiable since it is the list-decoding radius of our base code building block; ρ should be set as small as possible so that (4) can be derived from (1). The proof that (3)⇒(4) requires $\rho'' \geq \rho_{\mathsf{base}}^{\frac{t}{2}+o(1)}$ (the exponent is improved to $t + o(1)$ by working with wide replacement product walks [Ta-17]). In other words, the relationship between ρ_{base} and ρ'' is the same as the relationship between the distance of the base code and the distance of the random walk XOR code. This means that if we use a base code which is efficiently list-decodable for all $\rho_{\mathsf{base}} \geq \varepsilon_{\mathsf{base}}$ (such as the code from [GR08]), *and* if we could derive (3) from (1) with $\rho \approx \rho''$, then we would have given an explicit construction of a binary code with efficient decoder almost achieving list-decoding capacity.[3] Unfortunately, both of the steps (1)⇒(2) and (2)⇒(3) from [AJQ+20] and subsequent work incur loss. The loss in the (1)⇒(2) step is quadratic, $\rho' \geq \sqrt{\rho}$ is required. The loss in the (2)⇒(3) step is much larger, requiring $\rho'' = (\rho')^{o(1)}$. Our main technical contribution is a new analysis for the (2)⇒(3) step which incurs essentially no loss.

A Closer Look at the (2)⇒(3) Step From Prior Works. The key to proving (2)⇒(3) is to show that with non-negligible probability, the slice (S, σ) will be such that

$$\Delta^{(S,\sigma)} := \mathbb{E}_{\vec{a}\sim \mathsf{RW}_A^t}\left[\tilde{\mathbb{E}}^{(S,\sigma)}\left[(-1)^{\vec{\alpha}\oplus \mathbf{x}_{\vec{a}}}\right] - \prod_{i=1}^{t} \tilde{\mathbb{E}}^{(S,\sigma)}\left[(-1)^{\alpha_i \oplus \mathbf{x}_{a_i}}\right]\right]$$

is small, where the first pseudoexpectation is over $\vec{\alpha} \sim \mathcal{O}^{(S,\sigma)}(\vec{a})$, and where the $i-$th pseudoexpectation in the product is over $\vec{\alpha}_i \sim \mathcal{O}^{(S,\sigma)}(a_i)$. In words, we must show that with very high probability over $\vec{a} \sim \mathsf{RW}_A^t$, the distributions $\mathcal{O}^{(S,\sigma)}(\vec{a})$ and $\prod_i \mathcal{O}^{(S,\sigma)}(a_i)$ are close with respect to the "RW XOR test" which, given $\vec{\alpha}$, outputs $(-1)^{\vec{\alpha}\oplus \mathbf{x}_{\vec{a}}}$. If $\Delta^{(S,\sigma)}$ is not small, then it must be that significant correlations exist, on average, between the bits output by $\mathcal{O}^{(S,\sigma)}(\vec{a})$. Following the ideas of [AKK+08], it is possible to show that if the output bits of $\mathcal{O}^{(S,\sigma)}(\vec{a})$ are correlated with good probability over $\vec{a} \sim \mathsf{RW}_A^t$ *and* if A is a good expander, then the output bits of $\mathcal{O}^{(S,\sigma)}(\vec{a})$ will also be correlated with good probability over $\vec{a} \sim A^t$. Essentially this means that (S, σ) is a bad slice, since it is precisely the slice's job to remove such global correlations. Specifically, it is shown in [BRS11] that if (S, σ) is a bad slice, then (S', σ') will be a much better slice with good probability over $S' \supset S$ such that $|S'| = |S| + 1$, and $\sigma' \sim \mathcal{O}^{(S,\sigma)}(S')$. It can be shown that such significant improvement cannot be the norm, from which it follows that most slices are good.

The Idea Behind our Improved (2)⇒(3) Step. The setting considered in [AKK+08, BRS11] is not quite the same as our setting. These works consider expander edges rather than longer walks and it is shown that if $\mathcal{O}^{(S,\sigma)}\big(\{a, a'\}\big)$ is far from $\mathcal{O}^{(S,\sigma)}(a) \times \mathcal{O}^{(S,\sigma)}(a')$ in statistical distance for a random edge $a \sim a'$, then these distributions are also far for independent $a, a' \sim A$. This step incurs an additive

[3] "Almost" because of slight suboptimality of Ta-Shma's code which carries over also to our (3)⇒(4) step.

loss of λ, the expansion of A, which means the conclusion that the distributions are far for independent $a, a' \sim A$ (which is needed in order to show that (S', σ') will be a drastically improved slice) can only be obtained if the distributions are at least $\lambda-$far for a random edge $a \sim a'$. The decoding algorithm in [AJQ+20] (and subsequent work) extends the main theorem from [BRS11] to longer random walks using a hybrid argument, and inherits the requirement that the initial statistical distance must be at least $t\lambda$. Thus, they are only able to show that $\Delta^{(S,\sigma)} \leq t\lambda$ holds for most (S, σ) (we are oversimplifying; their algorithm uses several additional ideas and does slightly better).

Our starting point is that if $\Delta^{(S,\sigma)}$ is not small, then it means that the RW XOR test distinguishes $\mathcal{O}^{(S,\sigma)}(\vec{a})$ and $\prod_i \mathcal{O}^{(S,\sigma)}(a_i)$ for an average $\vec{a} \sim \mathsf{RW}_A^t$. Since random walks on expanders are good parity samplers, perhaps we could expect the distinguishing probability of the RW XOR test to decay like $\lambda^{t/2+o(1)}$ rather than like $t\lambda$. We prove this by showing that the "random walks on expanders are good parity samplers" theorem extends to pseudodistributions. The result is that we are able to show that $\Delta^{(S,\sigma)} \leq \lambda^{t/2+o(1)}$ with non-negligible probability over (S, σ); a substantial improvement over the bound obtained in prior work.

2 Technical Overview

In order to demonstrate our techniques, in this section we prove the key lemmas needed to give an improved list decoding algorithm for the random walk XOR code. Our main theorem will follow from the analogous result for the wide replacement walk XOR code. We begin this section with a short preliminaries section where we introduce just the concepts needed for the proofs in this section. Another, more substantial preliminaries section will follow this one where we will introduce the rest of the background material needed for this paper.

2.1 Preliminaries

Notation. For a distribution $\mathcal{D}$, we write $x \sim \mathcal{D}$ to mean that $x \in \mathrm{Supp}(\mathcal{D})$ is drawn according to $\mathcal{D}$. For a set S, we write $x \sim S$ instead of $x \sim \mathsf{Unif}(S)$. Vector norms in this work refer always to the ℓ_2-norm. For an integer $r \in \mathbb{N}$, we write $[r]$ as shorthand for $\{1, \ldots, r\}$.

Basic Statistics. For a real-valued random variable X, we write $\mathsf{Var}(\mathsf{X})$ for the variance of X, namely $\mathsf{Var}(\mathsf{X}) = \mathbb{E}[\mathsf{X}^2] - \mathbb{E}[\mathsf{X}]^2$. We will use that variance is non-increasing under conditioning. So if X and Y are jointly distributed real-valued random variables, then $\mathsf{Var}(\mathsf{X}) \geq \mathbb{E}_{y\sim\mathsf{Y}}\big[\mathsf{Var}(\mathsf{X}|y)\big]$. This follows from Jensen's inequality:

$$\begin{aligned}\mathsf{Var}(\mathsf{X}) - \mathbb{E}_{y\sim\mathsf{Y}}\big[\mathsf{Var}(\mathsf{X}|y)\big] &= \mathbb{E}[\mathsf{X}^2] - \mathbb{E}[\mathsf{X}]^2 - \mathbb{E}_{y\sim\mathsf{Y}}\big[\mathbb{E}[\mathsf{X}^2|y] - \mathbb{E}[\mathsf{X}|y]^2\big] \\ &= \mathbb{E}_{y\sim\mathsf{Y}}\big[\mathbb{E}[\mathsf{X}|y]^2\big] - \mathbb{E}_{y\sim\mathsf{Y}}\big[\mathbb{E}[\mathsf{X}|y]\big]^2 \geq 0\end{aligned}$$

For jointly distributed real-valued random variables X and Y, we write $\mathsf{Cov}(\mathsf{X}, \mathsf{Y})$ for the covariance of X and Y: $\mathsf{Cov}(\mathsf{X}, \mathsf{Y}) = \mathbb{E}[\mathsf{XY}] - \mathbb{E}[\mathsf{X}]\mathbb{E}[\mathsf{Y}]$. The following claim was proved in [BRS11] (Lemma C.2); we include a proof in Appendix A.

Claim 1. *Suppose* X *and* Y *are jointly distributed real-valued random variables, and moreover that* X *is supported on a set of size* 2*. Then*

$$\mathsf{Cov}(\mathsf{X}, \mathsf{Y})^2/\mathsf{Var}(\mathsf{X}) = \mathsf{Var}(\mathsf{Y}) - \mathbb{E}_{x\sim\mathsf{X}}\big[\mathsf{Var}(\mathsf{Y}|x)\big].$$

Random Walks on Graphs. Let A be the vertex set of a graph. Given $a, a' \in A$, we write $a \sim a'$ if a and a' are connected by an edge. For $a \in A$, let $N(a) \subset A$ denote the *neighborhood* of A, *i.e.*, $N(a) := \{a' \in A : a \sim a'\}$. For an integer $d \geq 1$, we say that A is $d-$regular if $|N(a)| = d$ for all $a \in A$. For an integer $k \geq 1$, let

$$\mathsf{RW}_A^k := \{(a_1, \ldots, a_k) \in A^k : a_i \sim a_{i+1} \; \forall \, i = 1, \ldots, k-1\}$$

denote the set of $k-$length random walks in A. Similarly, for $a \in A$, $\mathsf{RW}_A^k(a)$ is the set of $k-$length random walks in A which begin at a, so $\mathsf{RW}_A^k(a) := \{(a_1, \ldots, a_k) \in \mathsf{RW}_A^k : a_1 = a\}$. We will often view RW_A^k as a distribution, where $(a_1, \ldots, a_k) \sim \mathsf{RW}_A^k$ means that $a_1 \sim A$ is drawn uniformly and then $a_{i+1} \sim N(a_i)$ is drawn for $i = 1, \ldots, k-1$. In this section, we write RW^k instead of RW_A^k, since the graph A will not change.

Expander Graphs. The *expansion* of a graph is the second largest eigenvalue of the graph's adjacency matrix,[4] *i.e.*,

$$\lambda := \max_{x, y \perp \mathbb{1}} \frac{|\langle x, My \rangle|}{|x||y|},$$

where the max is over all nonzero $x, y \in \mathbb{R}^{|A|} - \{0\}$ which are perpendicular to the all 1s vector $\mathbb{1}$. The expander mixing lemma is recovered from this definition for any $f, g : A \to \mathbb{R}$ by setting $x, y \in \mathbb{R}^{|A|}$ to be $x_a = f(a) - \mu_f$ and $y_a = g(a) - \mu_g$.

Claim 2 (Expander Mixing Lemma). *If A is a $\lambda-$*expander *then for all $f, g : A \to \mathbb{R}$,*

$$\left| \mathbb{E}_{a \sim a'} \big[f(a) \cdot g(a') \big] - \mu_f \mu_g \right| \leq \lambda \tau_f \tau_g,$$

where μ_f and τ_f are the expectation and standard deviation of the random variable $f(a)$ (namely, $\mu_f = \mathbb{E}_a \big[f(a) \big]$ and $\tau_f^2 + \mu_f^2 = \mathbb{E}_a \big[f(a)^2 \big]$, and similarly for μ_g and τ_g).

We also will need the following "high-dimensional" version of the EML.

Claim 3 (EML for Vector-Valued Functions). *Suppose A is a $\lambda-$expander, and let $f, g : A \to \mathbb{R}^N$ be vector-valued functions defined on A. We have*

$$\left| \mathbb{E}_{a \sim a'} \big[\langle f(a), g(a') \rangle \big] - \langle \vec{\mu}_f, \vec{\mu}_g \rangle \right| \leq \lambda \tau_f \tau_g,$$

where $\vec{\mu}_f := \mathbb{E}_a \big[f(a) \big]$, and $\tau_f \in \mathbb{R}$ is such that $\tau_f^2 + |\vec{\mu}_f|^2 = \mathbb{E}_a \big[|f(a)|^2 \big]$ (and similarly for $\vec{\mu}_g$, τ_g).

Proof. For $i = 1, \ldots, N$, let $f_i, g_i : A \to \mathbb{R}$ be the coordinate functions of f and g, respectively. Note that $\tau_f^2 = \sum_i \tau_{f_i}^2$. We have

$$\begin{aligned} \left| \mathbb{E}_{a \sim a'} \big[\langle f(a), g(a') \rangle \big] - \langle \vec{\mu}_f, \vec{\mu}_g \rangle \right| &\leq \sum_{i=1}^N \left| \mathbb{E}_{a \sim a'} \big[f_i(a) \cdot g_i(a') \big] - \mu_{f_i} \mu_{g_i} \right| \leq \lambda \sum_{i=1}^N \tau_{f_i} \tau_{g_i} \\ &\leq \lambda \cdot \sqrt{\sum_i \tau_{f_i}^2} \cdot \sqrt{\sum_i \tau_{g_i}^2} = \lambda \tau_f \tau_g, \end{aligned}$$

by the triangle inequality, Claim 2 (applied to the coordinate functions), and Cauchy-Schwarz. □

[4] The *adjacency matrix* of the graph A is $M \in \{0,1\}^{|A| \times |A|}$, where $M(a, a') = 1$ iff $a \sim a'$.

2.2 Bias Amplification via Expander Random Walks

In order to get a sense of our techniques, we begin with a short proof that random walks on expanders are good parity samplers.

Claim 4. *Let A be a regular $\lambda-$expander, let $\mathbf{x} : A \to \{0,1\}$ be such that $\left|\mathbb{E}_a\left[(-1)^{\mathbf{x}_a}\right]\right| \leq \sqrt{\lambda}$. Then for all $k \in \mathbb{N}$,*

$$\mathbb{E}_{\vec{a}\sim \mathsf{RW}^k}\left[(-1)^{\mathbf{x}_{a_1}\oplus\cdots\oplus\mathbf{x}_{a_k}}\right] \leq \frac{1}{2}(4\lambda)^{\frac{k}{2}}.$$

Proof. For $k \in \mathbb{N}$, define the function $g_k : A \to \mathbb{R}$ via

$$g_k(a) := \mathbb{E}_{\vec{a}\sim \mathsf{RW}^k(a)}\left[(-1)^{\mathbf{x}_{a_1}\oplus\cdots\oplus\mathbf{x}_{a_k}}\right].$$

Define the statistics $\varepsilon_k := \left|\mathbb{E}_a[g_k(a)]\right|$ and τ_k such that $\varepsilon_k^2 + \tau_k^2 = \mathbb{E}_a\left[g_k(a)^2\right]$. We prove by induction that for all $k \in \mathbb{N}$,

$$\varepsilon_k \leq \frac{1}{2}(4\lambda)^{\frac{k}{2}}; \text{ and } \tau_k \leq (4\lambda)^{\frac{k-1}{2}}.$$

This proves the claim because it gives the desired bound on ε_k. For the base case, note $\varepsilon_1 \leq \sqrt{\lambda}$ holds by hypothesis and $\tau_1 \leq 1$ is trivial. The key point for the induction step is that for $k \geq 2$,

$$g_k(a) = (-1)^{\mathbf{x}_a} \cdot \mathbb{E}_{a'\sim N(a)}\left[g_{k-1}(a')\right].$$

This lets us bound ε_k and τ_k in terms of lower order statistics using the EML (Claim 2):

$$\cdot\ \varepsilon_k = \left|\mathbb{E}_a[g_k(a)]\right| = \left|\mathbb{E}_{a\sim a'}\left[(-1)^{\mathbf{x}_a}\cdot g_{k-1}(a')\right]\right| \leq \sqrt{\lambda}\varepsilon_{k-1} + \lambda\tau_{k-1};$$

$$\cdot\ \tau_k^2 \leq \varepsilon_k^2 + \tau_k^2 = \mathbb{E}_a\left[g_k(a)^2\right] = \mathbb{E}_a\left[\mathbb{E}_{\alpha\sim\mathcal{D}(a)}\left[(-1)^{\alpha}\right]^2 \cdot \mathbb{E}_{a'\sim N(a)}\left[g_{k-1}(a')\right]^2\right]$$

$$\leq \mathbb{E}_a\left[\mathbb{E}_{a'\sim N(a)}[g_{k-1}(a')]^2\right] = \mathbb{E}_{a'\sim_{A^2}a''}\left[g_{k-1}(a')\cdot g_{k-1}(a'')\right] \leq \varepsilon_{k-1}^2 + \lambda^2\tau_{k-1}^2,$$

where $a' \sim_{A^2} a''$ indicates that (a', a'') is a uniform edge in A^2 (a λ^2-expander). We have used that the distribution which draws $a \sim A$, $a', a'' \sim N(a)$ and outputs (a', a'') is identical to the uniform edge distribution on A^2. Plugging in the inductive hypothesis into the right hand sides of the above bounds and simplifying proves the induction step. □

2.3 Bias Amplification for Good Pseudodistributions via Expander RWs

We now demonstrate our new technique by proving that random walks on expander graphs are good parity samplers for "good" pseudodistributions. For a pseudodistribution to be "good", it must satisfy the technical requirements specified below. We will show later on that if $\mathcal{O}$ is the pseudodistribution obtained by solving the SDP of the decoding algorithm, then the pseudodistribution $\mathcal{O}^{(S,\sigma)}$ will be good with non-negligible probability over the slice (S, σ). This will require adopting the argument of [BRS11] to our modified rounding technique. We handle this, and other issues in Section 2.4. For the remainder of this section, we fix a good pseudodistribution $\mathcal{O}$ on $\{0,1\}^A$, where A is a $\lambda-$expander. We will specify what it means for $\mathcal{O}$ to be good below, after we set some notation.

- **The Random Variables $\{\mathsf{Y}_{\vec{a}}\}$:** For $\vec{a} \in \mathsf{RW}^k$, with $k \leq t$, let $\mathsf{Y}_{\vec{a}}$ be the distribution on $\{\pm 1\}$ which draws $\vec{\alpha} \sim \mathcal{O}(\vec{a})$ and outputs $(-1)^{\vec{\alpha}\oplus\mathbf{x}_{\vec{a}}}$. Note that the $\{\mathsf{Y}_{\vec{a}}\}$ are pairwise jointly distributed as long as $\mathcal{O}$ supports queries of size $2t$: given $\vec{a}, \vec{a}' \in \mathsf{RW}^{\leq t}$, $(\mathsf{Y}_{\vec{a}}, \mathsf{Y}_{\vec{a}'})$ is the distribution which draws $(\vec{\alpha}, \vec{\alpha}') \sim \mathcal{O}(\vec{a}\cup\vec{a}')$ and outputs $\left((-1)^{\vec{\alpha}\oplus\mathbf{x}_{\vec{a}}}, (-1)^{\vec{\alpha}'\oplus\mathbf{x}_{\vec{a}'}}\right)$.

- **The Covariance Vectors $\{\hat{\mathbf{v}}_{\vec{a}}\}$ and $\{\mathbf{v}_{\vec{a}}\}$:** Since the $\{\mathsf{Y}_{\vec{a}}\}$ are pairwise jointly distributed, we define their covariances $\mathsf{Cov}(\mathsf{Y}_{\vec{a}}, \mathsf{Y}_{\vec{a}'}) = \tilde{\mathbb{E}}\big[\mathsf{Y}_{\vec{a}}\mathsf{Y}_{\vec{a}'}\big] - \tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}]\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}'}]$. As covariance matrices are positive semidefinite, there exist real vectors $\{\hat{\mathbf{v}}_{\vec{a}}\}$ such that $\langle \hat{\mathbf{v}}_{\vec{a}}, \hat{\mathbf{v}}_{\vec{a}'}\rangle = \mathsf{Cov}(\mathsf{Y}_{\vec{a}}, \mathsf{Y}_{\vec{a}'})$ for all $\vec{a}, \vec{a}' \in \mathsf{RW}^{\leq t}$. For $\vec{a} \in \mathsf{RW}^{\leq t}$, let $\mathbf{v}_{\vec{a}}$ be the vector $\hat{\mathbf{v}}_{\vec{a}}$ with one extra coordinate which is equal to $\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}]$. Note for all $\vec{a}, \vec{a}' \in \mathsf{RW}^{\leq t}$,

$$\tilde{\mathbb{E}}\big[\mathsf{Y}_{\vec{a}}\mathsf{Y}_{\vec{a}'}\big] = \mathsf{Cov}(\mathsf{Y}_{\vec{a}}, \mathsf{Y}_{\vec{a}'}) + \tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}]\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}'}] = \langle \hat{\mathbf{v}}_{\vec{a}}, \hat{\mathbf{v}}_{\vec{a}'}\rangle + \tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}]\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}'}] = \langle \mathbf{v}_{\vec{a}}, \mathbf{v}_{\vec{a}'}\rangle.$$

- **The Statistics $\{\varepsilon_k, \mu_k, \tau_k\}$:** For $k \leq t$, define $g_k : A \to \mathbb{R}$ by $g_k(a) = \mathbb{E}_{\vec{a}\sim\mathsf{RW}^k(a)}\big[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}]\big]$, and let $\varepsilon_k := \big|\mathbb{E}_a[g_k(a)]\big|$. Similarly, for $a \in A$, define the vector $\mathbf{w}_k(a) := \mathbb{E}_{\vec{a}\sim\mathsf{RW}^k(a)}[\mathbf{v}_{\vec{a}}]$. Let $\mu_k := \big|\mathbb{E}_a[\mathbf{w}_k(a)]\big|$ and let τ_k be such that $\mu_k^2 + \tau_k^2 = \mathbb{E}_a\big[|\mathbf{w}_k(a)|^2\big]$.

- **Good Pseudodistribution:** Fix $\delta > 0$ such that $\delta \leq \frac{1}{9}(9\lambda)^{t-1}$. We say that a $\mathcal{O}$ is a *good pseudodistribution* if it supports queries of size $2t$, and if for all $k, k' \leq t$:

$$\mathbb{E}_{\substack{\vec{a}\sim\mathsf{RW}^k \\ \vec{a}'\sim\mathsf{RW}^{k'}}}\big[|\langle \hat{\mathbf{v}}_{\vec{a}}, \hat{\mathbf{v}}_{\vec{a}'}\rangle|\big] \leq \delta.$$

Lemma 1. *Let A be a $\lambda-$expander, and suppose $\mathcal{O}$ is a good pseudodistribution on $\{0,1\}^A$ which supports queries of size $2t$ and for which $\big|\mathbb{E}_a\big[\tilde{\mathbb{E}}[\mathsf{Y}_a]\big]\big| \leq \sqrt{\lambda}$. Then*

$$\left|\mathbb{E}_{\vec{a}\sim\mathsf{RW}^t}\left[\tilde{\mathbb{E}}\big[\mathsf{Y}_{\vec{a}}\big]\right]\right| \leq \frac{1}{3}\cdot(9\lambda)^{\frac{t}{2}}.$$

Proof. We will use induction to show that for all $k = 1, \ldots, t$:

$$\varepsilon_k \leq \frac{1}{3}\cdot(9\lambda)^{\frac{k}{2}}; \text{ and } \tau_k \leq (9\lambda)^{\frac{k-1}{2}}.$$

This proves the lemma since it gives the desired bound on ε_t. For the base case, $\varepsilon_1 \leq \sqrt{\lambda}$ holds by assumption, and $\tau_1 \leq 1$ is trivial. For the induction step, we will show that for all $k \geq 2$:

(i) $\varepsilon_k \leq \delta + \sqrt{\lambda}\varepsilon_{k-1} + \lambda\tau_{k-1}$;

(ii) $\tau_k \leq \sqrt{\delta} + \varepsilon_{k-1} + \lambda\tau_{k-1}$.

Invoking the induction hypothesis on the right hand side of (i) gives

$$\varepsilon_k \leq (9\lambda)^{\frac{k}{2}}\cdot\left[\frac{\delta}{(9\lambda)^{\frac{k}{2}}} + \frac{1}{9} + \frac{1}{9}\right] \leq \frac{1}{3}\cdot(9\lambda)^{\frac{k}{2}},$$

since $\delta \leq \frac{1}{9}\cdot(9\lambda)^{\frac{k}{2}}$. Similarly, invoking the induction hypothesis on the right hand side of (ii) gives

$$\tau_k \leq (9\lambda)^{\frac{k-1}{2}}\cdot\left[\frac{\sqrt{\delta}}{(9\lambda)^{\frac{k-1}{2}}} + \frac{1}{3} + \frac{\sqrt{\lambda}}{3}\right] \leq (9\lambda)^{\frac{k-1}{2}},$$

since $\delta \leq \frac{1}{9}\cdot(9\lambda)^{k-1}$. Thus, it remains to establish the bounds in (i) and (ii).

(i) $-$ **Bounding ε_k:** We have

$$\begin{aligned}\varepsilon_k &= \left|\mathbb{E}_{\vec{a}\sim\mathsf{RW}^k}\left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}]\right]\right| = \left|\mathbb{E}_{\substack{a\sim a'\\ \vec{a}'\sim\mathsf{RW}^{k-1}(a')}}\left[\tilde{\mathbb{E}}[\mathsf{Y}_a\mathsf{Y}_{\vec{a}'}]\right]\right| \leq \left|\mathbb{E}_{\substack{a\sim A\\ \vec{a}'\sim\mathsf{RW}^{k-1}}}\left[\tilde{\mathbb{E}}[\mathsf{Y}_a\mathsf{Y}_{\vec{a}'}]\right]\right| + \lambda\tau_{k-1}\\ &\leq \mathbb{E}_{\substack{a\sim A\\ \vec{a}'\sim\mathsf{RW}^{k-1}}}\left[\left|\langle\hat{\mathbf{v}}_a,\hat{\mathbf{v}}_{\vec{a}'}\rangle\right|\right] + \left|\mathbb{E}_a\left[\tilde{\mathbb{E}}[\mathsf{Y}_a]\right]\right|\cdot\left|\mathbb{E}_{\vec{a}'\sim\mathsf{RW}^{k-1}}\left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}'}]\right]\right| + \lambda\tau_{k-1}\\ &\leq \delta + \sqrt{\lambda}\varepsilon_{k-1} + \lambda\tau_{k-1}\end{aligned}$$

The inequality on the first line is the EML, using the identity

$$\mathbb{E}_{\substack{a\sim a\\ \vec{a}'\sim\mathsf{RW}^{k-1}(a')}}\left[\tilde{\mathbb{E}}[\mathsf{Y}_a\mathsf{Y}_{\vec{a}'}]\right] = \mathbb{E}_{a\sim a'}\left[\langle\mathbf{v}_a,\mathbf{w}_{k-1}(a')\rangle\right].$$

The inequality on the second line follows from the triangle inequality and Jensen's inequality; the final inequality has used that $\mathcal{O}$ is a good pseudodistribution.

(ii) $-$ **Bounding τ_k:** We have

$$\begin{aligned}\tau_k^2 &\leq \mathbb{E}_{\substack{a\sim A\\ \vec{a},\vec{a}'\sim\mathsf{RW}^k(a)}}\left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}\mathsf{Y}_{\vec{a}'}]\right] = \mathbb{E}_{\substack{a\sim A\\ \vec{a},\vec{a}'\sim\mathsf{RW}^k(a)}}\left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}_{2:k}}\mathsf{Y}_{\vec{a}'_{2:k}}]\right] = \mathbb{E}_{\substack{a\sim_{A^2}a'\\ \vec{a}\sim\mathsf{RW}^{k-1}(a)\\ \vec{a}'\sim\mathsf{RW}^{k-1}(a')}}\left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}\mathsf{Y}_{\vec{a}'}]\right]\\ &\leq \mathbb{E}_{\vec{a},\vec{a}'\sim\mathsf{RW}^{k-1}}\left[\left|\langle\hat{\mathbf{v}}_{\vec{a}},\hat{\mathbf{v}}_{\vec{a}'}\rangle\right|\right] + \mathbb{E}_{a,a'\sim A}\left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}]\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}'}]\right] + \lambda^2\tau_{k-1}^2 \leq \delta + \varepsilon_{k-1}^2 + \lambda^2\tau_{k-1}^2,\end{aligned}$$

from which $\tau_k \leq \sqrt{\delta} + \varepsilon_{k-1} + \lambda\tau_{k-1}$ follows. The first bound is trivial since $\tau_k^2 \leq \mu_k^2 + \tau_k^2$; the last equality on the first line holds because $\vec{a}$ and $\vec{a}'$ share the same first vertex and so $\alpha_1 = \alpha_1'$ holds with probability 1 over $(\vec{\alpha},\vec{\alpha}') \sim \mathcal{O}(\vec{a}\cup\vec{a}')$; the inequality on the second line is the EML on A^2 (a λ^2-expander); the inequalities on the final line hold by the triangle inequality, Jensen's inequality and since $\mathcal{O}$ is a good pseudodistribution.

□

The Key Takeaway. So to summarize, the proof of Lemma 1 is very similar to the proof of Claim 4. For example, the identity

$$\mathbb{E}_{\vec{a}\sim\mathsf{RW}^k}\left[\tilde{\mathbb{E}}[(-1)^{\vec{\alpha}\oplus\mathbf{x}_{\vec{a}}}]\right] = \mathbb{E}_{\substack{a\sim a'\\ \vec{a}\sim\mathsf{RW}^{k-1}(a)}}\left[\tilde{\mathbb{E}}[\mathsf{Y}_a\mathsf{Y}_{\vec{a}}]\right] = \mathbb{E}_{\substack{a\sim a'\\ \vec{a}\sim\mathsf{RW}^{k-1}(a)}}\left[\langle\mathbf{v}_a,\mathbf{v}_{\vec{a}}\rangle\right] = \mathbb{E}_{a\sim a'}\left[\langle\mathbf{v}_a,\mathbf{w}_{k-1}(a')\rangle\right],$$

means that the quantity $\mathbb{E}_{\vec{a}\sim\mathsf{RW}^k}\left[\tilde{\mathbb{E}}[(-1)^{\vec{\alpha}\oplus\mathbf{x}_{\vec{a}}}]\right]$ is subject to bounds via the (high-dimensional) EML, just as the quantity $\mathbb{E}_{\vec{a}\sim\mathsf{RW}^k}\left[(-1)^{\mathbf{x}_{\vec{a}}}\right]$ was bounded using the (one-dimensional) EML in the proof of Claim 4. However, this method when applied to pseudoexpectations leaves behind some extra "casualty terms" of the form $\mathbb{E}_{a\sim A,\vec{a}\sim\mathsf{RW}^{k-1}}\left[\left|\langle\hat{\mathbf{v}}_a,\hat{\mathbf{v}}_{\vec{a}}\rangle\right|\right]$ which did not appear in the proof of Claim 4. These casualties are (absolute values of the) covariances between $\mathsf{Y}_{\vec{a}}$ and $\mathsf{Y}_{\vec{a}'}$ for independent $\vec{a},\vec{a}' \in \mathsf{RW}^{\leq t}$. These can be bounded using a version of the "correlation potential" argument from [BRS11], as we will now demonstrate.

2.4 Decoding the Random Walk XOR Code

Armed with Lemma 1, let us now step back and describe in more detail the rounding step of our decoding algorithm and its analysis. For this discussion, let us pick up the decoding algorithm after it

has already solved the SDP (on input $\tilde{\mathbf{y}} \in \{0,1\}^{\mathsf{RW}^t}$) and obtained a pseudodistribution $\mathcal{O}$ on $\{0,1\}^A$ such that (2) holds; namely so that

$$\mathbb{E}_{\vec{a}\sim\mathsf{RW}^t}\Big[\tilde{\mathbb{E}}\big[(-1)^{\vec{\alpha}\oplus\mathbf{x}_{\vec{a}}}\big]\Big] \geq \rho',$$

where $\rho' = \frac{2}{3}\cdot(9\lambda)^{\frac{t}{2}}$ is twice the bound obtained in Lemma 1, and $\mathbf{x} \in \mathsf{LIST}(\tilde{\mathbf{y}}, \rho)$. Our rounding method, described next, will probabilistically generate a list L of data which determines a conditional pseudodistribution $\mathcal{O}^{\mathsf{L}}$. The consistency conditions of $\mathcal{O}$ will ensure that for all $\vec{a} \in \mathsf{RW}^t$, the distribution which draws L and then draws and outputs $\vec{\alpha} \sim \mathcal{O}^{\mathsf{L}}(\vec{a})$ is identical to $\mathcal{O}(\vec{a})$. Thus, with probability at least $\rho'/2$ over L, we will have

$$\mathbb{E}_{\vec{a}\sim\mathsf{RW}^t}\Big[\tilde{\mathbb{E}}^{\mathsf{L}}\big[(-1)^{\vec{\alpha}\oplus\mathbf{x}_{\vec{a}}}\big]\Big] \geq \rho'/2.$$

We will prove that with probability at least $1-\rho'/4$ over L, $\mathcal{O}^{\mathsf{L}}$ will satisfy:

$$\mathbb{E}_{\substack{\vec{a}\sim\mathsf{RW}^k\\ \vec{a}'\sim\mathsf{RW}^{k'}}}\big[|\langle\hat{\mathbf{v}}_{\vec{a}}, \hat{\mathbf{v}}_{\vec{a}'}\rangle|\big] \leq \delta,$$

for all $k, k' \leq t$. It then follows from Lemma 1 that with probability at least $\rho'/4$ over L,

$$\mathbb{E}_{a\sim A}\Big[\tilde{\mathbb{E}}^{\mathsf{L}}\big[(-1)^{\alpha\oplus\mathbf{x}_a}\big]\Big] \geq \sqrt{\lambda}.$$

Since $\sqrt{\lambda} \geq 2\rho_{\mathsf{base}}$, this implies that with probability at least $\rho'/4$ over L, the decoding algorithm will output, with high probability, a string $\vec{\alpha} \in \{0,1\}^A$ such that $\Pr_{a\sim A}\big[\alpha = \mathbf{x}_a\big] \geq \frac{1+\sqrt{\lambda}}{2}$, which will allow recovering $\mathbf{x}$ using the list decoding algorithm of the base code.

Our Rounding Method. The rounding step of our decoding algorithm works as follows.

1. Draw $\mathbf{s} = (s_1, \ldots, s_t) \sim \{1, \ldots, r\}^t$ uniformly.
2. Draw a random subset $S \subset \mathsf{RW}^{\leq t}$ such that $\big|S \cap \mathsf{RW}^k\big| = s_k$ for all $k = 1, \ldots, t$.
3. Draw $\sigma \sim \mathcal{O}(S)$.
4. Output $\mathsf{L} = \big\{(\vec{a}, \oplus\sigma_{\vec{a}}) : \vec{a} \in S\big\}$, where $\oplus\sigma_{\vec{a}}$ denotes the bit $\sigma_{a_1} \oplus \cdots \oplus \sigma_{a_k} \in \{0,1\}$, where $(\sigma_{a_1}, \ldots, \sigma_{a_k})$ are the bits of σ which correspond to $\vec{a} = (a_1, \ldots, a_k)$.

So the rounding step outputs a "slice" $\mathsf{L} \subset \mathsf{RW}^{\leq t} \times \{0,1\}$ of size at most tr. Let $S \subset \mathsf{RW}^{\leq t}$ denote the projection of L onto the first coordinate; for $\vec{a} \in S$, we denote the element of L whose first coordinate is $\vec{a}$ as $(\vec{a}, \beta_{\vec{a}})$. Given L, the pseudodistribution $\mathcal{O}^{\mathsf{L}}$, on receiving a set $T \subset A$ such that $|S \cup T| \leq t(r+2)$, draws $\sigma \sim \mathcal{O}(S \cup T)$ such that $\oplus\sigma_{\vec{a}} = \beta_{\vec{a}}$ holds for all $\vec{a} \in S$, then outputs $\sigma_T \in \{0,1\}^T$. Note for all L drawn according to the rounding procedure, the pseudodistribution $\mathcal{O}^{\mathsf{L}}$ can be queried on any set of size at most $2t$. As discussed above, the following key lemma implies the the correctness of the decoding algorithm.

Lemma 2. *Assume $r \geq 4t^2/(\delta^2\rho')$. Then with probability at least $1-\rho'/4$ over L, both of the following hold for all $k, k' \in [t]$:*

$$\mathbb{E}_{\substack{\vec{a}\sim\mathsf{RW}^k\\ \vec{a}'\sim\mathsf{RW}^{k'}}}\big[|\langle\hat{\mathbf{v}}_{\vec{a}}, \hat{\mathbf{v}}_{\vec{a}'}\rangle|\big] \leq \delta,$$

where $\hat{\mathbf{v}}_{\vec{a}}$ is the notation from Section 2.3 instantiated with the pseudodistribution $\mathcal{O}^{\mathsf{L}}$.

Proof. Let $\mathcal{L}$ denote the slice distribution. We must show that with probability $1 - \rho'/4$ over $\mathsf{L} \sim \mathcal{L}$, we have

$$\mathbb{E}_{\substack{\vec{a}\sim\mathsf{RW}^k \\ \vec{a}'\sim\mathsf{RW}^{k'}}} \left[|\langle \hat{\mathbf{v}}^{\mathsf{L}}_{\vec{a}}, \hat{\mathbf{v}}^{\mathsf{L}}_{\vec{a}'} \rangle| \right] \leq \delta,$$

for all $k, k' \in [t]$ where $\hat{\mathbf{v}}^{\mathsf{L}}_{\vec{a}}$ refers to $\hat{\mathbf{v}}_{\vec{a}}$ from Section 2.3 instantiated with the pseudodistribution $\mathcal{O}^{\mathsf{L}}$. In this proof we are explicit about L because we will have to consider these random variables for different slices. Recall $\langle \hat{\mathbf{v}}^{\mathsf{L}}_{a}, \hat{\mathbf{v}}^{\mathsf{L}}_{\vec{a}} \rangle = \mathsf{Cov}(\mathsf{Y}^{\mathsf{L}}_{a}, \mathsf{Y}^{\mathsf{L}}_{\vec{a}})$, where $\mathsf{Y}^{\mathsf{L}}_{\vec{a}}$ is the random variable which draws $\vec{\alpha} \sim \mathcal{O}^{\mathsf{L}}(\vec{a})$ and outputs $(-1)^{\vec{\alpha} \oplus \mathbf{x}_{\vec{a}}}$. By Claim 1 we have

$$\mathbb{E}_{\substack{\vec{a}\sim\mathsf{RW}^k \\ \vec{a}'\sim\mathsf{RW}^{k'}}} \left[|\langle \hat{\mathbf{v}}^{\mathsf{L}}_{\vec{a}}, \hat{\mathbf{v}}^{\mathsf{L}}_{\vec{a}'} \rangle| \right]^2 \leq \mathbb{E}_{\substack{\vec{a}\sim\mathsf{RW}^k \\ \vec{a}'\sim\mathsf{RW}^{k'}}} \left[\mathsf{Cov}(\mathsf{Y}^{\mathsf{L}}_{\vec{a}}, \mathsf{Y}^{\mathsf{L}}_{\vec{a}'})^2 \right] \leq \mathbb{E}_{\vec{a}\sim\mathsf{RW}^k} \left[\mathsf{Var}(\mathsf{Y}^{\mathsf{L}}_{\vec{a}}) - \mathbb{E}_{\substack{\vec{a}'\sim\mathsf{RW}^{k'} \\ \beta\sim\mathsf{Y}^{\mathsf{L}}_{\vec{a}'}}} \left[\mathsf{Var}(\mathsf{Y}^{\mathsf{L}'}_{\vec{a}}) \right] \right],$$

where $\mathsf{L}' = \mathsf{L} \cup \{(\vec{a}', \beta)\}$. We say that L' is a *k'−th increment* of L since it is obtained from L by adding a single element $(\vec{a}', \beta) \in \mathsf{RW}^{k'} \times \{0,1\}$. Let $\mathcal{L}^{k'}(\mathsf{L})$ denote the distribution which outputs a random slice which is a k'−th increment of L. So $\mathcal{L}^{k'}(\mathsf{L})$ draws $\vec{a}' \sim \mathsf{RW}^{k'}$, $\beta \sim \mathsf{Y}^{\mathsf{L}}_{\vec{a}'}$ and outputs $\mathsf{L}' = \mathsf{L} \cup \{(\vec{a}', \beta)\}$. For $k \in [t]$, define the potential $\Phi_k(\mathsf{L}) := \mathbb{E}_{\vec{a}\sim\mathsf{RW}^k} \left[\mathsf{Var}(\mathsf{Y}^{\mathsf{L}}_{\vec{a}}) \right]$. These notations simplify the above, we now have

$$\mathbb{E}_{\substack{\vec{a}\sim\mathsf{RW}^k \\ \vec{a}'\sim\mathsf{RW}^{k'}}} \left[|\langle \hat{\mathbf{v}}^{\mathsf{L}}_{\vec{a}}, \hat{\mathbf{v}}^{\mathsf{L}}_{\vec{a}'} \rangle| \right]^2 \leq \Phi_k(\mathsf{L}) - \mathbb{E}_{\mathsf{L}'\sim\mathcal{L}^{k'}(\mathsf{L})} \left[\Phi_k(\mathsf{L}') \right].$$

Thus, by the union bound and Markov's inequality, in order to prove Lemma 2, it suffices to show that for all $k, k' = 1, \ldots, t$,

$$\mathbb{E}_{\mathsf{L}\sim\mathcal{L}} \left[\Phi_k(\mathsf{L}) - \mathbb{E}_{\mathsf{L}'\sim\mathcal{L}^{k'}(\mathsf{L})} \left[\Phi_k(\mathsf{L}') \right] \right] \leq \frac{\delta^2 \rho'}{4t^2}. \tag{$\dagger$}$$

Now recall that $\mathcal{L}$ first draws $\mathbf{s} = (s_1, \ldots, s_t) \sim [r]^t$ and then outputs a random slice L subject to $\left| \mathsf{L} \cap (\mathsf{RW}^k \times \{0,1\}) \right| = s_k$ for all $k = 1, \ldots, t$. Let $\mathcal{L}_{\mathbf{s}}$ be the distribution which outputs a random slice subject to satisfying this size condition. So $\mathcal{L}$ draws $\mathbf{s} \sim [r]^t$ and outputs a sample from $\mathcal{L}_{\mathbf{s}}$, while the distribution which draws $\mathsf{L} \sim \mathcal{L}$ and outputs a sample from $\mathcal{L}^{k'}(\mathsf{L})$ is identical to the distribution which draws $\mathbf{s} \sim [r]^t$ and outputs a sample from $\mathcal{L}_{\mathbf{s}'}$ where $\mathbf{s}'$ is the k'-th increment of $\mathbf{s}$, namely, $s'_{k'} = s_{k'} + 1$ while $s'_j = s_j$ for all $j \neq k'$. Let $\Phi_k(\mathbf{s}) := \mathbb{E}_{\mathsf{L}\sim\mathcal{L}_{\mathbf{s}}} \left[\Phi_k(\mathsf{L}) \right]$. Then the left hand side of ($\dagger$) becomes $\mathbb{E}_{\mathbf{s}\sim[r]^t} \left[\Phi_k(\mathbf{s}) - \Phi_k(\mathbf{s}') \right]$, where $\mathbf{s}'$ denotes the k'−th increment of $\mathbf{s}$. But for all $s_1, \ldots, s_{k-1}, s_{k+1}, \ldots, s_t \in [r]$, we have

$$1 \geq \Phi_k(\mathbf{s}_1) \geq \Phi_k(\mathbf{s}_2) \geq \cdots \geq \Phi_k(\mathbf{s}_r) \geq 0,$$

where $\mathbf{s}_v$ denotes $(s_1, \ldots, s_{k-1}, v, s_{k+1}, \ldots, s_r) \in [r]^t$. This is because variance is non-increasing under conditioning. It follows that the LHS of ($\dagger$) is at most $1/r \leq \delta^2 \rho'/(4t^2)$. □

3 Preliminaries

In this section we give the rest of preliminaries which we have not already given in Section 2.1.

Basic Notations. For subsets $S, T \subset \mathcal{U}$ of some universe $\mathcal{U}$, we write $S \oplus T$ for the "exclusive OR" of S and T: $S \oplus T = (S \cup T) \setminus (S \cap T)$. For a string $\vec{\alpha} = (\alpha_1, \ldots, \alpha_k) \in \{0,1\}^k$, we will frequently write $(-1)^{\vec{\alpha}}$ as shorthand for $(-1)^{\alpha_1 \oplus \cdots \oplus \alpha_k}$.

3.1 List-Decodable Codes

Definition 1. *Let $\{\mathcal{C}_k\}$ be a family of binary linear codes with $\mathcal{C}_k : \{0,1\}^k \to \{0,1\}^n$, and let $\rho > 0$. We say that $\{\mathcal{C}_k\}$ is $\rho-$*list-decodeable *if there exists an efficient[5] family of algorithms $\{\mathcal{D}_k\}_k$ which takes $\tilde{y} \in \{0,1\}^n$ as input and outputs $\mathsf{LIST}(\tilde{y}, \rho) \subset \{0,1\}^k$, where recall*

$$\mathsf{LIST}(\tilde{y}, \rho) := \Big\{ m \in \{0,1\}^k : \Delta\big(\tilde{y}, \mathcal{C}_k(m)\big) \leq \frac{1-\rho}{2} \Big\}.$$

The following is proved in [GR08].

Proposition 1 (**The Base Code.**). *For any $\varepsilon_{\mathsf{base}} > 0$, there exists an explicit family $\{\mathcal{C}_k^{\mathsf{base}}\}_k$ of binary linear codes with $\mathcal{C}_k^{\mathsf{base}} : \{0,1\}^k \to \{0,1\}^n$ which has distance $\frac{1-\varepsilon_{\mathsf{base}}}{2}$, rate $\Omega\big(\varepsilon_{\mathsf{base}}^3\big)$ and which is $\rho_{\mathsf{base}}-$list-decodable for all $\rho_{\mathsf{base}} \geq \varepsilon_{\mathsf{base}}$.*

3.2 The Lasserre Semidefinite Programming Hierarchy

The Lasserre hierarchy is a convenient framework for wielding the SDP algorithmic machinery to optimize constraint satisfaction problems (CSPs). Practically speaking, the Lasserre setup it is part of an SDP program, and instantiating the hierarchy involves completing the setup to a full SDP by incorporating some "problem specific" components. Our decoding algorithm will be looking for an assignment $\mathbf{x} : A \to \{0,1\}$ and so we will set up a SDP which has variables $\{\mathbf{z}_{S,\sigma}\}_{S,\sigma}$, and which includes the following constraints:

• **Variables:** There is one variable $\mathbf{z}_{S,\sigma}$ for each $S \subset A$ of size $|S| \leq r$ and $\sigma \in \{0,1\}^S$.

• **Lasserre Constraints:** There are four types of Lasserre constraints.

(i) $|\mathbf{z}_{\emptyset,\emptyset}|^2 = 1$;

(ii) $|\mathbf{z}_{a,0}|^2 + |\mathbf{z}_{a,1}|^2 = 1 \; \forall \, a \in A$;

(iii) $\langle \mathbf{z}_{S,\sigma}, \mathbf{z}_{T,\tau} \rangle = 0 \; \forall \, S, T \subset A$ such that $S \cap T \neq \emptyset$ and $\sigma|_{S\cap T} \neq \tau|_{S\cap T}$;

(iv) $\langle \mathbf{z}_{S,\sigma}, \mathbf{z}_{T,\tau} \rangle = \langle \mathbf{z}_{S',\sigma'}, \mathbf{z}_{T',\tau'} \rangle \; \forall \, S, T, S', T' \subset A$ s.t. $S \cup T = S' \cup T'$ and $\sigma \cup \tau = \sigma' \cup \tau'$.

Our decoding algorithm will "complete" this setup to a full SDP by specifying a convex objective to minimize and by adding some additional constraints. When all is said an done, the final SDP will be solvable to within additive accuracy $\eta > 0$ in time $\mathsf{poly}\big(2^r|A|^r, \log(1/\eta)\big)$ since there are (fewer than) $2^r|A|^r$ variables. Upon solving the SDP, one obtains vectors $\{\mathbf{z}_{S,\sigma}\}_{S,\sigma}$ which (nearly) optimize the objective. Working with the SDP solution directly, while possible, carries significant notational overhead which can obstruct the intuition of the rest of the algorithm. For this reason, convenient language has been developed for reasoning about Lasserre-type SDP solutions. We will also use this language and here we briefly discuss how to convert the above into a nicer form.

The starting point is that the Lasserre constraints imply that for all $S \subset A$ of size $|S| \leq r$, $\sum_\sigma |\mathbf{z}_{S,\sigma}|^2 = 1$, and so for all S, the SDP solution describes a distribution $\mathcal{O}(S)$ on $\{0,1\}^S$ which outputs $\sigma \in \{0,1\}^S$ with probability $|\mathbf{z}_{S,\sigma}|^2$ (this was the notation used in Sections 1.2 and 2). It follows from the Lasserre constraints that for $S \subset T$, the S marginal of $\mathcal{O}(T)$ is identical to $\mathcal{O}(S)$.

[5] In this work, we consider an algorithm efficient if it runs in expected polynomial time in n; we allow the exponent to depend on ρ.

Definition 2 (Pseudodistributions). *A* level r Lasserre pseudodistribution *on* $\{0,1\}^A$ *(or just a pseudodistribution) is an oracle* $\mathcal{O}$ *with the following syntax and and which satisfies the following consistency requirement.*

- **Syntax:** *On receiving a set* $S \subset A$ *of size* $|S| \leq r$*,* $\mathcal{O}$ *probabilistically generates and returns the string* $\sigma \in \{0,1\}^S$*; we let* $\mathcal{O}(S)$ *denote the distribution that* $\mathcal{O}$ *uses to respond to the query* S*.*

- **Consistency:** *for all* $S \subset T$*, the* $S-$*marginal of* $\mathcal{O}(T)$ *is identical to* $\mathcal{O}(S)$*.*

For any $S \subset A$ *such that* $|S| < r$ *and any* $\sigma \in \{0,1\}^S$*, we define the* conditional pseudodistribution $\mathcal{O}^{(S,\sigma)}$ *by letting* $\mathcal{O}^{(S,\sigma)}(T)$ *be the distribution which draws* $\tau \sim \mathcal{O}(S \cup T)$ *such that* $\tau|_S = \sigma$ *and outputs* $\tau|_T \in \{0,1\}^T$*. Note* $\mathcal{O}^{(S,\sigma)}(T)$ *is defined for all* $T \subset A$ *such that* $|S \cup T| \leq r$*, and so* $\mathcal{O}^{(S,\sigma)}$ *is a level* $r - |S|$ *pseudodistribution.*

In order to maximally simplify the notation of the SDP, it is common to define the remainder of the SDP in terms of the pseudodistribution $\mathcal{O}$ rather than the variables $\{\mathbf{z}_{(S,\sigma)}\}_{S,\sigma}$. Thus, when using the Lasserre hierarchy, we will wind up with an SDP that looks like this:

- **Minimize:** $\Phi(\mathcal{O})$;

- **Problem-Specific Constraints:** $\Gamma_i(\mathcal{O}) \leq a_i$ for $i = 1, 2, \ldots$;

- **Lasserre Constraints:** $\mathcal{O}$ is a level r Lasserre pseudodistribution;

where Φ and the Γ_i are convex functions. Convexity in this context means that for any two pseudodistributions $\mathcal{O}$ and $\mathcal{O}'$, and any $\gamma \in (0,1)$, $\Phi\big(\gamma\mathcal{O} + (1-\gamma)\mathcal{O}'\big) \leq \gamma\Phi(\mathcal{O}) + (1-\gamma)\Phi(\mathcal{O}')$, where $\gamma\mathcal{O} + (1-\gamma)\mathcal{O}'$ is the pseudodistribution which, for $S \subset A$ of size $|S| \leq r$ outputs $\sigma \in \{0,1\}^S$ with probability $\gamma\Pr\big[\mathcal{O}(S) = \sigma\big] + (1-\gamma)\Pr\big[\mathcal{O}'(S) = \sigma\big]$.

3.3 The Wide Replacement Product

Ta-Shma's code used a special graph product called the *wide replacement product* to combine two expanders. This method was introduced in [BT11] where it was used to get almost Ramanujan expander graphs via the zig-zag product. We describe this method as it applies to combining the two specific graphs used in the construction.

Cayley Graphs Given a finite group G and a subset $U \subseteq G$, the Cayley graph $\mathrm{Cayley}(G,U)$ has vertex set G with $g \sim g'$ iff $g^{-1}g' \in U$. Note that $\mathrm{Cayley}(G,U)$ is $|U|-$regular; additionally, if U is closed under inversion, then $\mathrm{Cayley}(G,U)$ is undirected. Cayley graphs play a key role in many explicit constructions of expander graphs. Both the outer and inner graphs used in this work are explicit expander constructions based on Cayley graphs.

3.3.1 The Outer Graph

Rotation Maps and Local Invertibility. In any $d-$regular, undirected graph $G = (V,E)$, we can define the *rotation map* $\phi : V \times [d] \to V$ defined via $\phi : (v,i) \mapsto v'$, where $v' \in N(v)$ is the $i-$th neighbor of v. Note that if $v' = \phi(v,i)$ then there must exist some $i' \in [d]$ such that $v = \phi(v',i')$, since G is undirected. In this case we say that i' is the *inverse of* i *with respect to the vertex* v, since

$v = \phi\big(\phi(v,i),i'\big)$. We say that G is *locally invertible* if for all i there exists i' such that $v = \phi\big(\phi(v,i),i'\big)$ holds for all $v \in V$. In Cayley graphs, the rotation map is simply multiplication: $\phi(g,u) = gu$. This means that undirected Cayley graphs are locally invertible, since for all $g \in G$ and $u \in U$, $g = \phi\big(\phi(g,u)u^{-1}\big)$.

For our outer graph A we use the explicit construction of [Alo21] which, though not a Cayley graph, is locally invertible as it is a combination of Cayley graphs.

Proposition 2 (The Outer Graph). *For all integers $n, d \in \mathbb{N}$ there is an explicit construction of an undirected, $d-$regular, locally invertible graph A with $n \cdot (1 + o_n(1))$ vertices and expansion $\lambda_A \leq \frac{8}{\sqrt{d}}$.*

3.3.2 The Inner Graph

For our inner graph B we use the explicit construction of [AGHP92].

Proposition 3 (The Inner Graph). *For all integers $r, \ell \in \mathbb{N}$ such that $\ell \leq r/2$, there exists an explicit construction of an undirected $2^{2\ell}-$regular Cayley graph B over $\mathbb{F}_2^r$ with expansion $\lambda_B \leq (r-1)2^{-\ell}$.*

The Shifted Neighborhood Distribution. We will introduce parameters $s, m \in \mathbb{N}$ such that $r = ms$ so that our inner graph B is a Cayley graph on $\mathbb{F}_2^{ms}$. We will view elements $b \in B$ as $s-$tuples $b = \big(b[1], \ldots, b[s]\big) \in (\mathbb{F}_2^m)^s$ and we write $\hat{b} \in \mathbb{F}_2^m$ for the first coordinate of b, namely $\hat{b} = b[1]$. We define the "shift operator" via $\mathsf{shift} : \big(b[1], \ldots, b[s]\big) \mapsto \big(b[2], \ldots, b[s], b[1]\big)$. For $b \in B$, we define the *shifted neighborhood distribution of* b, denoted $\tilde{N}(b)$ as the distribution which draws $b' \sim N(b)$ and outputs $\mathsf{shift}(b')$. The shifted neighborhood distribution was introduced in [RR22] to simplify the notations of Ta-Shma's construction, which draws a random walk $(b_1, \ldots, b_t) \sim \mathsf{RW}_B^t$ and then used the indices $\big(b_1[1], b_2[2], \ldots, b_t[t \bmod s]\big) \in (\mathbb{F}_2^m)^t$. This is equivalent to drawing a *shifted random walk* $(b_1, \ldots, b_t) \sim \widetilde{\mathsf{RW}}_B^t$ (*i.e.*, drawing $b_1 \sim B$ and $b_{i+1} \sim \tilde{N}(b_i)$ for $i = 1, \ldots, t-1$), and then outputting $(\hat{b}_1, \ldots, \hat{b}_t) \in (\mathbb{F}_2^m)^t$. It is easy to see that the expansion of B is not affected by using the shifted neighborhood distribution instead of the original neighborhood distribution:

$$\left|\mathbb{E}_{\substack{b\sim B\\ b'\sim\tilde{N}(b)}}\big[f(b)\cdot g(b')\big] - \mu_f\mu_g\right| = \left|\mathbb{E}_{\substack{b\sim B\\ b'\sim N(b)}}\big[f(b)\cdot \tilde{g}(b')\big] - \mu_f\mu_{\tilde{g}}\right| \leq \lambda\sigma_f\sigma_{\tilde{g}} = \lambda\sigma_f\sigma_g,$$

where $\tilde{g} = g \circ \mathsf{shift}$; clearly $(\mu_{\tilde{g}}, \sigma_{\tilde{g}}) = (\mu_g, \sigma_g)$. Note that by replacing the normal neighborhood distribution with the shifted neighborhood distribution, we turn B into an undirected graph. To fix this, we can define the *reverse shifted neighborhood distribution* $\tilde{N}^{-1}(b)$ which draws $b' \sim N(b)$ and outputs $\mathsf{shift}^{-1}(b')$, where shift^{-1} shifts the coordinates of b in the opposite direction to shift. The key point we will use is that the following distributions are identical: 1) draw $b \sim B$, $b' \sim \tilde{N}(b)$ and output $(b, b') \in B^2$; 2) draw $b' \sim B$, $b \sim \tilde{N}^{-1}(b')$ and output $(b, b') \in B^2$. The following easy claim will be useful.

Claim 5. *For $k \leq s$, the distribution which draws $(b_1, \ldots, b_k) \sim \widetilde{\mathsf{RW}}_B^k$ and outputs $(\hat{b}_1, \ldots, \hat{b}_k) \in \mathbb{F}_2^{mk}$ is identical to $\mathsf{Unif}(\mathbb{F}_2^{mk})$.*

Proof. Let $\mathcal{D}_k$ be the distribution which draws $(b_1, \ldots, b_k) \sim \widetilde{\mathsf{RW}}_B^k$ and outputs $(\hat{b}_1, \ldots, \hat{b}_k) \in \mathbb{F}_2^{mk}$. We must show that $\mathcal{D}_k \equiv \mathsf{Unif}(\mathbb{F}_2^{mk})$ for all $k \leq s$. It suffices to prove this for $k = s$, since when $k < s$, $\mathcal{D}_k$ is identical to the distribution which draws $(\hat{b}_1, \ldots, \hat{b}_s) \sim \mathcal{D}_s$ and outputs $(\hat{b}_1, \ldots, \hat{b}_k)$. As B is a Cayley graph on $\mathbb{F}_2^{ms}$, there exists a subset $U \subset \mathbb{F}_2^{ms}$ such that, for all $b \in B$, the neighborhood

distribution $N(b)$ draws $u \sim U$ and outputs $b+u$. Thus, $\mathcal{D}_s$ draws $b = \big(b[1],\ldots,b[s]\big) \sim \mathbb{F}^{ms}$ and $u_1,\ldots,u_{s-1} \sim U$, and outputs

$$(\hat{b}_1,\ldots,\hat{b}_s) = \Big(b[i] + \sum_{j<i} u_j[i-j+1]\Big)_{i=1,\ldots,s} \in \mathbb{F}_2^{ms}.$$

Uniformity of $(\hat{b}_1,\ldots,\hat{b}_s)$ follows from the uniformity of the $b[i] \sim \mathbb{F}_2^m$. □

3.3.3 The s-wide Replacement Product

Let A and B be the outer and inner graphs described above, respectively. The wide replacement product is a distribution parametrized by integers $s,t \in \mathbb{N}$ which uses a $t-$step walk on B to derive and output a walk $(a_0,\ldots,a_t) \in \mathsf{RW}_A^{t+1}$. We denote the $s-$wide replacement product distribution as $s\text{-}\mathsf{WRW}_{A,B}^t$. Since A and B will not change, we omit them from the syntax and we relocate the s, writing WRW_s^t instead of $s\text{-}\mathsf{WRW}_{A,B}^t$.[6] In order to ensure compatibility between A and B, we set $d = 2^m$, where d is the degree of A and B is a Cayley graph on $\mathbb{F}_2^{ms}$. This allows viewing $b = \big(b[1],\ldots,b[s]\big) \in B$ as an element of $[d]^s$. For $a \in A$ and $b \in B$, we interpret $\phi(a,\hat{b})$ as the $j-$th neighbor of a where $j \in [d]$ is the value which corresponds to $\hat{b} = b[1] \in \mathbb{F}_2^m \simeq [d]$.

The Distribution WRW_s^t. With the above setup in mind, the distribution WRW_s^t draws $a \sim A$, $(b_1,\ldots,b_t) \sim \widetilde{\mathsf{RW}}_B^t$ and outputs $(a_0,\ldots,a_t) \in A^{t+1}$, where $a_0 = a$ and $a_{i+1} = \phi(a_i,\hat{b}_i)$ for $i = 0,\ldots,t-1$. For $a \in A$, $\mathsf{WRW}_s^t(a)$ is the distribution WRW_s^t conditioned on $a_0 = a$; for $(a,b) \in A \times B$, $\mathsf{WRW}_s^t(a,b)$ is WRW_s^t conditioned on $(a_0,b_1) = (a,b)$.

4 The Code

We now describe the binary code $\{\mathcal{C}_k\}_k$ of Ta-Shma [Ta-17], which almost achieves the Gilbert-Varshimov bound. Fix $k \in \mathbb{N}$ and $\varepsilon > 0$. The construction of $\mathcal{C}_k$ uses the following building blocks.

- **The Base Code:** Let $\mathcal{C}_k^{\mathsf{base}} : \{0,1\}^k \to \{0,1\}^n$ be an explicit code of distance $\frac{1-\varepsilon_{\mathsf{base}}}{2}$, which is $\rho_{\mathsf{base}}-$list decodable for all $\rho_{\mathsf{base}} \geq \varepsilon_{\mathsf{base}}$. We use the construction of Proposition 1, which has rate $\Omega\big(\varepsilon_{\mathsf{base}}^3\big)$.
- **The Outer Graph:** Let A be the undirected d_A-regular graph with expansion λ_A. We use the construction of Proposition 2, so that $\lambda_A \leq 8/\sqrt{d_A}$ and $|A| = n \cdot \big(1+o_n(1)\big)$.
- **The Inner Graph:** Let B be a d_B-regular Cayley graph over $\mathbb{F}_2^r$ with expansion λ_B. We use the construction of Proposition 3 so that $\lambda_B = (r-1)\cdot 2^{-\ell}$ and $d_B = 2^{2\ell}$ for integers $\ell, r \in \mathbb{N}$ such that $\ell \leq r/2$.

The building blocks carry several parameters which we now connect. In order to set up the $s-$wide replacement product, define additional parameters $s,m \in \mathbb{N}$ such that $r = ms$, and let $d_A = 2^m$, so $B \simeq [d_A]^s$. It will be important for our analysis to have $\lambda_A \leq \lambda_B^2$; in order to arrange this, set $m = s$ and $\ell = s/5$. This gives

$$\lambda_A \leq \frac{8}{\sqrt{d_A}} = 8 \cdot 2^{-m/2} = \frac{8}{2^{\ell/2}} \cdot 2^{-2\ell} \leq (ms-1)^2 \cdot 2^{-2\ell} = \lambda_B^2,$$

[6] We intend WRW to stand for "wide replacement walk".

where the final inequality holds whenever $s \geq 2$. We will also require $\varepsilon_{\mathsf{base}} \leq \lambda_B/2$ which we ensure by setting $\varepsilon_{\mathsf{base}} = \frac{s^2-1}{2} \cdot 2^{-s/5}$. At this point, all parameters so far have been defined in terms of s. Note that our setup allows us to use B to take $s-$wide replacement walks in A. We now describe the code.

The Encoding Algorithm $\mathcal{C}_k$. On input $\mathsf{msg} \in \{0,1\}^k$, and given $\varepsilon, \eta > 0$, $\mathcal{C}_k$ works as follows.

1. Set $s, t \in \mathbb{N}$ so that $\eta s \geq 60 \log s$ and $t \geq s^2$ so that $(2\lambda_B)^{t(1-4/s)} \leq \varepsilon$.
2. Compute $\mathcal{C}_k^{\mathsf{base}}(\mathsf{msg}) \in \{0,1\}^n$, and define $\mathbf{x} \in \{0,1\}^A$ by setting
$$\mathbf{x}_a = \begin{cases} \mathcal{C}_k^{\mathsf{base}}(\mathsf{msg})_i, & a = \iota(i) \\ 0, & \text{otherwise} \end{cases}$$
where $\iota : [n] \hookrightarrow A$ is some fixed embedding.
3. Compute $\mathbf{y} \in \{0,1\}^{\mathsf{WRW}_s^t}$ by setting $\mathbf{y}_{\vec{a}} = \mathbf{x}_{a_0} \oplus \cdots \oplus \mathbf{x}_{a_t}$ for $\vec{a} = (a_0, \ldots, a_t) \in \mathsf{WRW}_s^t$.
4. Output $\mathbf{y}$.

Proposition 4. *The code $\{\mathcal{C}_k\}_k$ has distance at least $\frac{1-\varepsilon}{2}$, and rate $\Omega(\varepsilon^{2+\eta})$.*

Proof. The rate of $\mathcal{C}_k$ is
$$\mathsf{Rate}_k = \frac{k}{|\mathsf{WRW}_s^t|} \geq \frac{k}{|A|} \cdot \frac{1}{|B|} \cdot \frac{1}{d_B^{t-1}} = \Omega(\varepsilon_{\mathsf{base}}^3) \cdot 2^{-s^2} \cdot d_B^{-(t-1)} = \Omega\big(s^{-6} \cdot 2^{-s^2}\big) \cdot d_B^{-(t-1)}.$$
To bound the bias of $\mathcal{C}_k$, we use the following lemma which was proved in [Ta-17]. This result is also implied by Claim 7, since the constant pseudodistribution is technically good. The reader can also see also [RR22] for a proof of precisely this result which uses language similar to the rest of this paper.

Lemma 3 (Bias Reduction of Wide Replacement Product Walks). *Let integers $s, t \in \mathbb{N}$ and graphs A and B be as above; so in particular A and B are λ_A and λ_B expanders with $\lambda_A \leq \lambda_B^2$. Let $\mathbf{x} : A \to \{0,1\}$ be any function such that $\big|\mathbb{E}_a\big[(-1)^{\mathbf{x}_a}\big]\big| \leq \lambda_B$. Then*
$$\left|\mathbb{E}_{\vec{a} \sim \mathsf{WRW}_s^t}\left[(-1)^{\mathbf{x}_{\vec{a}}}\right]\right| \leq (2\lambda_B)^{t(1-4/s)}.$$

Note that the function $\mathbf{x} : A \to \{0,1\}$ defined in the second step of $\mathcal{C}_k$ satisfies
$$\left|\mathbb{E}_a\big[(-1)^{\mathbf{x}_a}\big]\right| \leq 2 \cdot \left|\mathbb{E}_{i \sim [n]}\big[(-1)^{\mathcal{C}_k^{\mathsf{base}}(\mathsf{msg})_i}\big]\right| \leq 2\varepsilon_{\mathsf{base}} \leq \lambda_B,$$
and so Lemma 3 ensures that $\mathsf{Bias}_k \leq (2\lambda_B)^{t(1-4/s)}$. Putting the calculations of Rank_k and Bias_k together and using $\lambda_B = (s^2-1)/\sqrt{d_B}$ gives
$$\mathsf{Rate}_k = \Omega\Big(s^{-6} \cdot (s^2-1)^{-2t} \cdot 2^{-2t-s^2+2s/5} \cdot (2\lambda_B)^{8t/s}\Big) \cdot \mathsf{Bias}_k^2 = \Omega\Big(s^{-5t} \cdot (2\lambda_B)^{8t/s}\Big) \cdot \mathsf{Bias}_k^2,$$
where the right most equality holds whenever $6 \log s \leq 2s/5$ (implied by $s \geq 100$) and $t \geq s^2$. Note, therefore, that for $\eta \in \big(0, 1/2\big)$, $\mathsf{Rate}_k = \Omega\big(\mathsf{Bias}_k^{2+\eta}\big)$ holds whenever $(2\lambda_B)^{t(\eta - 4\eta/s - 8/s)} \leq s^{-5t}$ which, if $\eta \geq 24/s$ is implied by $(2\lambda_B)^{\eta/2} \leq s^{-5}$. Finally, by plugging in $\lambda_B = (s^2-1) \cdot 2^{-s/5}$, we see that this holds whenever $\eta s \geq 60 \log s$.

So finally, let us prove the theorem. Suppose that we are given $\varepsilon > 0$ and $\eta \in \big(0, 1/2\big)$, and we want to construct $\mathcal{C}_k$ such that $\mathsf{Bias}_k \leq \varepsilon$ and $\mathsf{Rate}_k = \Omega\big(\mathsf{Bias}^{2+\eta}\big)$. We let $\mathcal{C}_k$ be the construction defined above with s chosen large enough so that $\eta s \geq 60 \log s$; this ensures $\mathsf{Rate}_k = \Omega\big(\mathsf{Bias}_k^{2+\eta}\big)$ as noticed above. Finally, let us choose t large enough so that $t \geq s^2$ and $(2\lambda_B)^{t(1-4/s)} \leq \varepsilon$; this ensures $\mathsf{Bias}_k \leq \varepsilon$, as desired. □

5 The List Decoding Algorithm

In this section, we describe the list-decoding algorithm. As already mentioned, this algorithm is essentially the same as the algorithms from [AJQ+20, JQST20, JST21] except for a modified rounding step.

The Decoding Algorithm $\mathcal{D}_k$. Let $\rho > 0$. On input $\tilde{\mathbf{y}} \in \{0,1\}^{\mathsf{WRW}_s^t}$, $\mathcal{D}_k$ does the following.

0. Set parameters $\delta = 2^{-14}\rho^8$, $r \geq \frac{16s^2t^2}{\delta^2\rho^4}$, and $\zeta = \frac{1}{16}\rho^8$.

1. Set up and solve the SDP to within accuracy ζ:
   - minimize $\mathbb{E}_{\vec{a},\vec{a}' \sim \mathsf{WRW}_s^t}\big[\tilde{\mathbb{E}}[(-1)^{\vec{\alpha}\oplus\vec{\alpha}'}]^2\big]$;
   - subject to $\mathbb{E}_{\vec{a}\sim\mathsf{WRW}_s^t}\big[\tilde{\mathbb{E}}[(-1)^{\tilde{\mathbf{y}}_{\vec{a}}\oplus\vec{\alpha}}]\big] \geq \rho$;
   - subject to $\mathcal{O}$ being a level $(rst+2)(s+t+1)$ Lasserre pseudodistribution.

2. Round:
   - Draw $\mathbf{S} = (\mathbf{S}_{(\ell,k)}) \sim [r]^{st}$ uniformly.
   - Draw a subset $S \subset \big\{(\vec{a},\vec{a}') \in \mathsf{RW}_A^\ell \times \mathsf{WRW}_s^k : a_\ell = a'_0\big\}$ randomly, subject to the requirement that $\big|S \cap (\mathsf{RW}_A^\ell \times \mathsf{WRW}_s^k)\big| = \mathbf{S}_{(\ell,k)}$ for all $\ell \in [s]$ and $k \in [t]$.
   - Draw $\sigma \sim \mathcal{O}(S)$, where $\mathcal{O}$ is the optimal pseudodistribution recovered in Step 1.
   - Output $\mathsf{L} = \big\{\big(\vec{a}\oplus\vec{a}', (\oplus\sigma_{\vec{a}})\oplus(\oplus\sigma_{\vec{a}'})\big) : (\vec{a},\vec{a}') \in S\big\}$, where $\oplus\sigma_{\vec{a}} = \sigma_{a_1}\oplus\cdots\oplus\sigma_{a_\ell} \in \{0,1\}$, where $(\sigma_{a_1},\ldots,\sigma_{a_\ell})$ are the bits of σ which correspond to $\vec{a} = (a_1,\ldots,a_\ell)$, and similarly for $\oplus\sigma_{\vec{a}'}$.

3. Compute $\tilde{\mathbf{x}} \in \{0,1\}^A$ as follows:
   - For any a which appears in any of the paths in S, set $\tilde{\mathbf{x}}_a = \sigma_a$;
   - For all other a, draw $\tilde{\mathbf{x}}_a \sim \mathcal{O}^{\mathsf{L}}(a)$, where $\mathcal{O}^{\mathsf{L}}$ is the conditional pseudodistribution on $\{0,1\}^A$ created from the pseudodistribution $\mathcal{O}$ obtaines in Step 1 and the set L produced in Step 2. So specifically, for $T \subset A$ such that $|S\cup T| \leq (rst+2)(s+t+1)$, $\mathcal{O}^{\mathsf{L}}$ draws $\sigma \sim \mathcal{O}(S\cup T)$ such that $(\oplus\sigma_{\vec{a}}) \oplus (\oplus\sigma_{\vec{a}'}) = \beta_{\vec{a},\vec{a}'}$ for all $(\vec{a},\vec{a}') \in S$, then outputs $\sigma_T \in \{0,1\}^T$.

4. Run the inner decoder:
   - Compute $\mathsf{L}_{\mathsf{base}} = \mathcal{D}_k^{\mathsf{base}}(\tilde{\mathbf{x}})$ and $\mathsf{L}'_{\mathsf{base}} = \mathcal{D}_k^{\mathsf{base}}(\tilde{\mathbf{x}}')$, where $\mathcal{D}_k^{\mathsf{base}}$ is the $\rho_{\mathsf{base}}-$list-decoding algorithm for the inner code $\mathcal{D}_k^{\mathsf{base}}$, and where $\tilde{\mathbf{x}}'$ is the opposite of $\tilde{\mathbf{x}}$ (*i.e.*, every bit is flipped);[7]
   - If $\mathsf{L}_{\mathsf{base}} \cup \mathsf{L}'_{\mathsf{base}} \neq \emptyset$, output a random element from either set, otherwise give no output.

Theorem 2 (Main). *The above algorithm $\mathcal{D}_k$ runs in time $k^{\mathsf{poly}(1/\rho)}$, and provides the following output guarantee: if $\tilde{\mathbf{y}} \in \{0,1\}^{\mathsf{WRW}_s^t}$ and $\mathsf{msg} \in \{0,1\}^k$ are such that $\Delta\big(\tilde{\mathbf{y}}, \mathcal{C}_k(\mathsf{msg})\big) \leq \frac{1-\rho}{2}$, for $\rho \geq 2\sqrt{\varepsilon}$, then $\mathcal{D}_k(\mathsf{msg})$ outputs msg with non-negligible probability.*[8]

[7] If $\mathcal{D}_k^{\mathsf{base}}$ is given a string $\mathbf{x} \in \{0,1\}^A$ as input, then it first recovers $\iota^{-1}(\mathbf{x}) \in \{0,1\}^n$ and ignores the bits of $\mathbf{x}$ corresponding to $a \notin \mathrm{Image}(\iota)$.

[8] Recall from Proposition 4 that the distance of $\mathcal{C}_k$ is at least $\frac{1-\varepsilon}{2}$.

Note that $\mathcal{D}_k(\tilde{\mathbf{y}})$ can be called repeatedly to recover the full list

$$\left\{\mathsf{msg} \in \{0,1\}^k : \Delta\big(\tilde{\mathbf{y}}, \mathcal{C}_k(\mathsf{msg})\big) \leq \frac{1-\rho}{2}\right\}$$

with high probability.

5.1 Analysis Overview

Here we give the high level proof of Theorem 2 by reducing it to the proof of two key lemmas, Lemmas 4 and 5, below. These lemmas in turn will be proved in the following sections. First note that the running time of $\mathcal{D}_k$ is dominated by Step 1, which runs in time $\mathsf{poly}\big(n^{(rst+2)(s+t+1)}, 1/\zeta\big) = k^{\mathsf{poly}(1/\rho)}$. So now, assume $\tilde{\mathbf{y}} \in \{0,1\}^{\mathsf{WRW}_s^t}$ and $\mathsf{msg} \in \{0,1\}^k$ are such that $\Delta\big(\tilde{\mathbf{y}}, \mathcal{C}_k(\mathsf{msg})\big) \leq \frac{1-\rho}{2}$. We must show that $\mathcal{D}_k(\tilde{\mathbf{y}})$ outputs msg with non-negligible probability. Define the set

$$\mathsf{LIST}(\tilde{\mathbf{y}}, \rho) := \left\{\mathbf{x} \in \{0,1\}^A : \mathbb{E}_{\vec{a} \sim \mathsf{WRW}_s^t}\left[(-1)^{\tilde{\mathbf{y}}_{\vec{a}} \oplus \mathbf{x}_{\vec{a}}}\right] \geq \rho\right\}.$$

This is an abuse of notation because technically this list should be a subset of $\{0,1\}^k$, but this list will be our main focus during this proof. Let $\mathbf{x} = \iota \circ \mathcal{C}_k(\mathsf{msg}) \in \{0,1\}^A$ and note that $\mathbf{x} \in \mathsf{LIST}(\tilde{\mathbf{y}}, \rho)$. Thus, it suffices to show that for all $\mathbf{x} \in \mathsf{LIST}(\tilde{\mathbf{y}}, \rho)$, the string $\tilde{\mathbf{x}} \in \{0,1\}^A$ computed during Step 3 of $\mathcal{D}(\tilde{\mathbf{y}})$ will, with non-negligible probability, be such that $\left|\mathbb{E}_a[(-1)^{\mathbf{x}_a \oplus \tilde{\mathbf{x}}_a}]\right| \geq 3\rho_{\mathsf{base}}/2$. Indeed, this in turn implies that $\left|\mathbb{E}_{i \sim [n]}[(-1)^{\mathbf{x}_{\iota(i)} \oplus \tilde{\mathbf{x}}_{\iota(i)}}]\right| \geq \rho_{\mathsf{base}}$, and so msg will appear either in $\mathsf{L}_{\mathsf{base}}$ or $\mathsf{L}'_{\mathsf{base}}$ since the inner code is $\rho_{\mathsf{base}}-$list decodeable.

Now, recall that $\mathcal{D}_k(\tilde{\mathbf{y}})$ produces $\tilde{\mathbf{x}}$ by drawing each bit independently from the size 1 marginals of the conditional pseudodistribution $\mathcal{O}^{\mathsf{L}}$, where L is produced in Step 2 via the rounding procedure. Therefore, by the Chernoff-Hoeffding inequality, $\mathbf{x} \oplus \tilde{\mathbf{x}}$ will be $3\rho_{\mathsf{base}}/2-$biased with high probability whenever

$$\left|\mathbb{E}_a\left[\tilde{\mathbb{E}}^{\mathsf{L}}[(-1)^{\mathbf{x}_a \oplus \alpha}]\right]\right| \geq 2\rho_{\mathsf{base}}, \tag{+}$$

where $\tilde{\mathbb{E}}^{\mathsf{L}}$ is the conditional pseudoexpectation corresponding to $\mathcal{O}^{\mathsf{L}}$. So in summary, we need to show that (+) holds for all $\mathbf{x} \in \mathsf{LIST}(\tilde{\mathbf{y}}, \rho)$. The following lemma was proved in [AJQ+20] (combination of Lemmas 6.3 and 6.4). We include a shortened version of their proof, converted to our language, in Appendix A.

Claim 6. *Let $\mathcal{O}$ be the pseudodistribution obtained during Step 1 of $\mathcal{D}_k(\tilde{\mathbf{y}})$. For all $\mathbf{x} \in \mathsf{LIST}(\tilde{\mathbf{y}}, \rho)$,*

$$\mathbb{E}_{\vec{a}, \vec{a}' \sim \mathsf{WRW}_s^t}\left[\tilde{\mathbb{E}}[(-1)^{\mathbf{x}_{\vec{a}} \oplus \vec{\alpha} \oplus \mathbf{x}_{\vec{a}'} \oplus \vec{\alpha}'}]\right] \geq \frac{1}{4}\rho^4. \tag{++}$$

So the core of our analysis involves deducing (+) from (++). This is accomplished using the following two lemmas which will be proved over the next two sections. Both lemmas involve the notion of a *good conditional pseudodistribution* which is a technical condition defined in Definition 3 in Section 5.2.

Lemma 4. *Assume $r \geq \frac{16 s^2 t^2}{\delta^2 \rho^4}$. Then with probability at least $1 - \frac{1}{16}\rho^4$, the slice L drawn during Step 2 of decoding is good.*

Lemma 5. *If $\mathcal{O}$ is a pseudodistribution on $\{0,1\}^A$ and L is a good slice such that*

$$\mathbb{E}_{\vec{a}, \vec{a}' \sim \mathsf{WRW}_s^t}\left[\tilde{\mathbb{E}}^{\mathsf{L}}[(-1)^{\mathbf{x}_{\vec{a}} \oplus \vec{\alpha} \oplus \mathbf{x}_{\vec{a}'} \oplus \vec{\alpha}'}]\right] \geq \frac{1}{8}\rho^4$$

holds, then (+) also holds.

These lemmas combine to complete the proof of Theorem 2 since it follows that if (++) holds for the pseudodistribution $\mathcal{O}$ recovered in Step 1, then with probability at least $\frac{1}{16}\rho^4$ over the slice drawn during Step 2, (+) holds as well. In Section 5.2 we define good and prove Lemma 4; Lemma 5 is proved in Section 5.3.

5.2 Good Pseudodistributions and Proof of Lemma 4

In this section we define what it means for a pseudodistribution on $\{0,1\}^A$ to be good, and we prove Lemma 4. We first set some notation.

- **The Graph Expansion Parameters:** The code uses two expander graphs B and A with expansions $\lambda_B = \lambda$ and $\lambda_A \le \lambda^2$, respectively, where $\lambda > 0$ is chosen so that $\lambda \le 2\rho_{\mathsf{base}}$, the list-decoding radius of the inner code. Additionally, the width s and length t of the replacement walks are such that $(2\lambda)^{t(1-4/s)} \le \frac{1}{4}\rho^2$ (since $\rho \ge 2\sqrt{\varepsilon}$).

- **The Random Variables $\{\mathsf{Y}_S\}$:** For $S = \{a_1, \dots, a_k\} \subset A$, with $|S| \le s+t+1$, let Y_S be the distribution on $\{\pm 1\}$ which draws $\sigma = (\sigma_1, \dots, \sigma_k) \sim \mathcal{O}^{\mathsf{L}}(S)$ and outputs $(-1)^{(\sigma_1 \oplus \mathbf{x}_{a_1}) \oplus \cdots \oplus (\sigma_k \oplus \mathbf{x}_{a_k})}$. Note that the $\{\mathsf{Y}_S\}$ are pairwise jointly distributed since $\mathcal{O}$ supports queries of size $2(s+t+1)$: given $S, S' \subset A$, $(\mathsf{Y}_S, \mathsf{Y}'_S)$ is the distribution which draws $(\sigma, \sigma') \sim \mathcal{O}^{\mathsf{L}}(S \cup S')$ and outputs

$$\left((-1)^{(\sigma_1 \oplus \mathbf{x}_{a_1}) \oplus \cdots \oplus (\sigma_k \oplus \mathbf{x}_{a_k})}, (-1)^{(\sigma'_1 \oplus \mathbf{x}_{a'_1}) \oplus \cdots \oplus (\sigma'_{k'} \oplus \mathbf{x}_{a'_{k'}})}\right).$$

  We will be particularly interested in Y_S for $S = \vec{a}$ for a wide random walk $\vec{a} \in \mathsf{WRW}_s^{\le t}$; this will be our main case of interest. However, we will also need to consider Y_S for $S = \vec{a} \oplus \vec{a}'$ for $\vec{a} \in \mathsf{RW}_A^{\ell}$ and $\vec{a}' \in \mathsf{WRW}_s^k$ for $\ell \le s$ and $k \le t$ such that $a_\ell = a'_k$.

- **The Covariance Vectors $\{\hat{\mathbf{v}}_S\}$ and $\{\mathbf{v}_S\}$:** Since the $\{\mathsf{Y}_S\}$ are pairwise jointly distributed, we can define their covariances $\mathsf{Cov}(\mathsf{Y}_S, \mathsf{Y}_{S'}) = \tilde{\mathbb{E}}\left[\mathsf{Y}_S \mathsf{Y}_{S'}\right] - \tilde{\mathbb{E}}[\mathsf{Y}_S]\tilde{\mathbb{E}}[\mathsf{Y}_{S'}]$. As covariance matrices are positive semi-definite, there exist real vectors $\{\hat{\mathbf{v}}_S\}$ such that $\langle \hat{\mathbf{v}}_S, \hat{\mathbf{v}}_{S'} \rangle = \mathsf{Cov}(\mathsf{Y}_S, \mathsf{Y}_{S'})$ for all $S, S' \subset A$ such that $|S|, |S'| \le s+t+1$. Let $\mathbf{v}_S$ be the vector $\hat{\mathbf{v}}_S$ with one extra coordinate which is equal to $\tilde{\mathbb{E}}[\mathsf{Y}_S]$. Thus, $\tilde{\mathbb{E}}\left[\mathsf{Y}_S \mathsf{Y}_{S'}\right] = \langle \mathbf{v}_S, \mathbf{v}_{S'} \rangle$ for all $S, S' \subset A$ such that $|S|, |S'| \le s+t+1$.

Definition 3 (Good Pseudodistributions). *Let $\delta > 0$ such that $\delta \le \frac{1}{64}(2\lambda)^{4t}$. We say that a level $2(s+t+1)$ Lasserre pseudodistribution on $\{0,1\}^A$ is* good *if for all $\ell, \ell' \in [s]$ and $k, k' \in [t]$, the following holds:*

$$\mathbb{E}_{\substack{\vec{a} \sim \mathsf{RW}_A^{\ell}, \vec{a}' \sim \mathsf{RW}_A^{\ell'} \\ \vec{a}'' \sim \mathsf{WRW}_s^k(a_\ell) \\ \vec{a}''' \sim \mathsf{WRW}_s^{k'}(a'_{\ell'})}} \left[\left| \langle \hat{\mathbf{v}}_{\vec{a} \oplus \vec{a}''}, \hat{\mathbf{v}}_{\vec{a}' \oplus \vec{a}'''} \rangle \right| \right] \le \delta,$$

where a_ℓ and $a'_{\ell'}$ denote, respectively, the final vertices in the random walks $\vec{a}$ and $\vec{a}'$.

Lemma 4 (Restated). *Assume $r \ge \frac{16 s^2 t^2}{\delta^2 \rho^4}$. Then with probability at least $1 - \frac{1}{16}\rho^4$, the slice L drawn during Step 2 of decoding is good.*

Proof. Let $\mathcal{L}$ denote the slice distribution. We must show that with probability $1 - \frac{1}{16}\rho^4$ over $\mathsf{L} \sim \mathcal{L}$, we have

$$\mathbb{E}(\mathsf{L}) := \mathbb{E}_{\vec{a}, \vec{a}', \vec{a}'', \vec{a}'''} \left[\left| \langle \hat{\mathbf{v}}^{\mathsf{L}}_{\vec{a} \oplus \vec{a}''}, \hat{\mathbf{v}}^{\mathsf{L}}_{\vec{a}' \oplus \vec{a}'''} \rangle \right| \right] \le \delta,$$

for all $\ell, \ell' \le s$ and $k, k' \le t$, where the expectation is over $\vec{a} \sim \mathsf{RW}_A^{\ell}$, $\vec{a}' \sim \mathsf{RW}_A^{\ell'}$, $\vec{a}'' \sim \mathsf{WRW}_s^k(a_\ell)$ and $\vec{a}''' \sim \mathsf{WRW}_s^{k'}(a'_{\ell'})$. Here $\hat{\mathbf{v}}^{\mathsf{L}}_{\vec{a}\oplus\vec{a}''}$ refers to $\hat{\mathbf{v}}_{\vec{a}\oplus\vec{a}''}$ from Section 2.3 instantiated with the pseudodistribution $\mathcal{O}^{\mathsf{L}}$. In this proof we are explicit about L because we will have to consider these random variables for different slices. Recall $\langle \hat{\mathbf{v}}^{\mathsf{L}}_{\vec{a}\oplus\vec{a}''}, \hat{\mathbf{v}}^{\mathsf{L}}_{\vec{a}'\oplus\vec{a}'''} \rangle = \mathsf{Cov}(\mathsf{Y}^{\mathsf{L}}_{\vec{a}\oplus\vec{a}''}, \mathsf{Y}^{\mathsf{L}}_{\vec{a}'\oplus\vec{a}'''})$, where $\mathsf{Y}^{\mathsf{L}}_{\vec{a}\oplus\vec{a}''}$ is the random variable which draws $(\vec{\alpha}, \vec{\alpha}'') \sim \mathcal{O}^{\mathsf{L}}(\vec{a} \oplus \vec{a}'')$ and outputs $(-1)^{\vec{\alpha}\oplus\mathbf{x}_{\vec{a}}\oplus\vec{\alpha}''\oplus\mathbf{x}_{\vec{a}''}}$. By Claim 1 we have

$$\mathbb{E}(\mathsf{L})^2 \le \mathbb{E}_{\vec{a},\vec{a}',\vec{a}'',\vec{a}'''}\left[\mathsf{Cov}(\mathsf{Y}^{\mathsf{L}}_{\vec{a}\oplus\vec{a}''}, \mathsf{Y}^{\mathsf{L}}_{\vec{a}'\oplus\vec{a}'''})^2\right] \le \mathbb{E}_{\vec{a},\vec{a}''}\left[\mathsf{Var}(\mathsf{Y}^{\mathsf{L}}_{\vec{a}\oplus\vec{a}''}) - \mathbb{E}_{\substack{\vec{a}',\vec{a}''' \\ \beta\sim\mathsf{Y}^{\mathsf{L}}_{\vec{a}\oplus\vec{a}''}}}\left[\mathsf{Var}(\mathsf{Y}^{\mathsf{L}'}_{\vec{a}\oplus\vec{a}''})\right]\right],$$

where $\mathsf{L}' = \mathsf{L} \cup \{(\vec{a}' \oplus \vec{a}''', \beta)\}$. We say that L' is an $(\ell', k')-$*th increment* of L since it is obtained from L by adding a single element $(\vec{a}' \oplus \vec{a}''', \beta) \in \mathsf{RW}_A^{\ell'} \times \mathsf{WRW}_s^{k'} \times \{0,1\}$ with $a'_{\ell'} = a'''_0$. Let $\mathcal{L}^{(\ell',k')}(\mathsf{L})$ denote the distribution which outputs a random slice which is an $(\ell', k')-$th increment of L. So $\mathcal{L}^{(\ell',k')}(\mathsf{L})$ draws $\vec{a}' \sim \mathsf{RW}_A^{\ell'}$, $\vec{a}''' \sim \mathsf{WRW}_s^{k'}(a_{\ell'})$ and $\beta \sim \mathsf{Y}^{\mathsf{L}}_{\vec{a}'\oplus\vec{a}'''}$ and outputs $\mathsf{L}' = \mathsf{L} \cup \{(\vec{a}' \oplus \vec{a}''', \beta)\}$. For $\ell \in [s]$ and $k \in [t]$, define the potential

$$\Phi_{(\ell,k)}(\mathsf{L}) := \mathbb{E}_{\substack{\vec{a}\sim\mathsf{RW}_A^{\ell} \\ \vec{a}''\sim\mathsf{WRW}_s^k(a_\ell)}}\left[\mathsf{Var}(\mathsf{Y}^{\mathsf{L}}_{\vec{a}\oplus\vec{a}''})\right].$$

These notations simplify the above, we now have $\mathbb{E}(\mathsf{L})^2 \le \Phi_{(\ell,k)}(\mathsf{L}) - \mathbb{E}_{\mathsf{L}'\sim\mathcal{L}^{(\ell',k')}(\mathsf{L})}\left[\Phi_{(\ell,k)}(\mathsf{L}')\right]$. Thus, by the union bound and Markov's inequality, in order to prove Lemma 4, it suffices to show that for all $\ell, \ell' \in [s]$ and $k, k' \in [t]$,

$$\mathbb{E}_{\mathsf{L}\sim\mathcal{L}}\left[\Phi_{(\ell,k)}(\mathsf{L}) - \mathbb{E}_{\mathsf{L}'\sim\mathcal{L}^{(\ell',k')}(\mathsf{L})}\left[\Phi_{(\ell,k)}(\mathsf{L}')\right]\right] \le \frac{\delta^2\rho^4}{16s^2t^2}. \tag{$\dagger$}$$

Now recall that $\mathcal{L}$ draws the tuple $\mathbf{S} = (\mathbf{S}_{(\ell,k)})_{\ell,k} \sim [r]^{st}$ and then outputs a random slice L subject to $\left|\mathsf{L} \cap (\mathsf{RW}_A^{\ell} \times \mathsf{WRW}_s^k \times \{0,1\})\right| = \mathbf{S}_{(\ell,k)}$ for all $\ell \in [s]$ and $k \in [t]$. Let $\mathcal{L}_{\mathbf{S}}$ be the distribution which outputs a random slice subject to satisfying this size condition. So $\mathcal{L}$ draws $\mathbf{S} \sim [r]^{st}$ and outputs a sample from $\mathcal{L}_{\mathbf{S}}$, while the distribution which draws $\mathsf{L} \sim \mathcal{L}$ and outputs a sample from $\mathcal{L}^{(\ell',k')}(\mathsf{L})$ is identical to the distribution which draws $\mathbf{S} \sim [r]^{st}$ and outputs a sample from $\mathcal{L}_{\mathbf{S}'}$ where $\mathbf{S}'$ is the (ℓ', k')-th increment of $\mathbf{S}$, namely, $\mathbf{S}'_{(\ell',k')} = \mathbf{S}_{(\ell',k')} + 1$ while all other indices of $\mathbf{S}$ and $\mathbf{S}'$ are equal. Let $\Phi_{(\ell,k)}(\mathbf{S}) := \mathbb{E}_{\mathsf{L}\sim\mathcal{L}_{\mathbf{S}}}\left[\Phi_{(\ell,k)}(\mathsf{L})\right]$. Then the LHS of ($\dagger$) becomes $\mathbb{E}_{\mathbf{S}\sim[r]^{st}}\left[\Phi_{(\ell,k)}(\mathbf{S}) - \Phi_{(\ell,k)}(\mathbf{S}')\right]$, where $\mathbf{S}'$ denotes the $(\ell', k')-$th increment of $\mathbf{S}$. But for all $\hat{\mathbf{S}} \in [r]^{st-1}$, we have

$$1 \ge \Phi_{(\ell,k)}(\mathbf{S}_1) \ge \Phi_{(\ell,k)}(\mathbf{S}_2) \ge \cdots \ge \Phi_{(\ell,k)}(\mathbf{S}_r) \ge 0,$$

where $\mathbf{S}_v \in [r]^{st}$ has the $(\ell', k')-$th coordinate equal v and all other coordinates equal $\hat{\mathbf{S}}$. This is because variance is non-increasing under conditioning. It follows then that the LHS of ($\dagger$) is at most $1/r \le \frac{\delta^2\rho^4}{16s^2t^2}$, as desired. □

5.3 Bias Amplification

In this section we prove Lemma 5 and show that wide replacement product walks are parity samplers for good pseudodistributions. Let us keep using the notations from the previous section, *i.e.*, the random variables $\{\mathsf{Y}_S\}$ and the vectors $\{\hat{\mathbf{v}}_S\}$ and $\{\mathbf{v}_S\}$, all instantiated using some good pseudodistribution $\mathcal{O}^{\mathsf{L}}$. Additionally, define the statistics $\{\mu_k, \tau_k, \varepsilon_k\}_k$ as follows. For $k \le t$, let the vector-valued function $\mathbf{w}_k$ on $A \times B$ be defined via

$$\mathbf{w}_k(a, b) := \mathbb{E}_{\vec{a}\sim\mathsf{WRW}_s^k(a,b)}[\mathbf{v}_{\vec{a}}],$$

and let $\mu_k := \left|\mathbb{E}_{a,b}[\mathbf{w}_k(a,b)]\right|$, and τ_k be such that $\mu_k^2 + \tau_k^2 = \mathbb{E}_{a,b}\left[|\mathbf{w}_k(a,b)|^2\right]$. Additionally, let $\varepsilon_k := \left|\mathbb{E}_{\vec{a}\sim\mathsf{WRW}_s^k}\left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}]\right]\right|$. In this section, all pseudoexpectations are for the pseudodistribution $\mathcal{O}^{\mathsf{L}}$; since there is no chance for confusion, we write $\tilde{\mathbb{E}}$ instead of $\tilde{\mathbb{E}}^{\mathsf{L}}$ to keep the notations simpler.

Lemma 5 (Restated). *If $\mathcal{O}^{\mathsf{L}}$ is a good level $2(s+t+1)$ pseudodistribution on $\{0,1\}^A$ such that $\left|\mathbb{E}_{a\sim A}\left[\tilde{\mathbb{E}}[\mathsf{Y}_a]\right]\right| \le \lambda$, then*

$$\mathbb{E}_{\vec{a},\vec{a}'\sim\mathsf{WRW}_s^t}\left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}\mathsf{Y}_{\vec{a}'}]\right] \le 2(2\lambda)^{2t(1-4/s)}.$$

Proof. Note $\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}\mathsf{Y}_{\vec{a}'}] = \langle \mathbf{v}_{\vec{a}}, \mathbf{v}_{\vec{a}'}\rangle = \langle \hat{\mathbf{v}}_{\vec{a}}, \hat{\mathbf{v}}_{\vec{a}'}\rangle + \tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}]\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}'}]$. Thus,

$$\left|\mathbb{E}_{\vec{a},\vec{a}'\sim\mathsf{WRW}_s^t}\left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}\mathsf{Y}_{\vec{a}'}]\right]\right| \le \mathbb{E}_{\vec{a},\vec{a}'\sim\mathsf{WRW}_s^t}\left[|\langle \hat{\mathbf{v}}_{\vec{a}}, \hat{\mathbf{v}}_{\vec{a}'}\rangle|\right] + \left|\mathbb{E}_{\vec{a}\sim\mathsf{WRW}_s^t}\left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}]\right]\right|^2 \le \delta + \varepsilon_t^2,$$

where the last inequality holds by definition of ε_t and because $\mathcal{O}^{\mathsf{L}}$ is good (using $\ell = \ell' = 1$ and $k = k' = t$). So it suffices to show that for all $k \le t$:

$$\varepsilon_k \le (2\lambda)^{k(1-4/s)}; \text{ and } \tau_k \le (2\lambda)^{(k-2)(1-4/s)}. \tag{$*$}$$

We prove this by induction. The following claim captures the core of the proof; we prove it in Section 6.

Claim 7. *Suppose $\mathcal{O}^{\mathsf{L}}$ is a good level $2(s+t+1)$ pseudodistribution on $\{0,1\}^A$ and $\left|\mathbb{E}_a\left[\tilde{\mathbb{E}}[\mathsf{Y}_a]\right]\right| \le \lambda$. Let ε_k and τ_k be as above. Then we have the following bounds.*

- **Part 1:** *When $k \le s$, we have $\varepsilon_k \le \sqrt{\delta} + \frac{1}{2}(2\lambda)^{k+1}$.*
- **Part 2:** *When $k \le s$, we have $\tau_k \le 3\sqrt{\delta} + 2(2\lambda)^{k-1}$.*
- **Part 3:** *When $k \ge s+1$, we have $\varepsilon_k \le \sqrt{\delta} + \frac{1}{2}(2\lambda)^s\left[\varepsilon_{k-s} + 3\tau_{k-s}\right]$.*
- **Part 4:** *When $k \ge s+1$, we have*

$$\tau_k^2 \le 2\sqrt{\delta} + (2\lambda)^{s-1}\left[\varepsilon_{k-s} + 3\tau_{k-s}\right]\varepsilon_{k-2} + 2\lambda^4(2\lambda)^{2(s-2)}\left[\varepsilon_{k-s} + 3\tau_{k-s}\right]^2 + \lambda^2\tau_{k-1}^2.$$

Establishing $(*)$ from Claim 7 is a straightforward induction. The base cases follow immediately from the bounds given in Parts 1 and 2. To prove the induction step, first note that the induction hypothesis implies that $(2\lambda)^{s-4}\left[\varepsilon_{k-s} + 3\tau_{k-s}\right] \le (2\lambda)^{k(1-4/s)} + 3(2\lambda)^{(k-2)(1-4/s)}$. Plugging this into the right side of Part 3 and simplifying proves the induction step for ε_k. Finally, plugging in the induction hypothesis into the right side of Part 4 bounds τ_k^2 by

$$(2\lambda)^{2(k-2)(1-4/s)}\left[\frac{2\sqrt{\delta}}{(2\lambda)^{2(k-2)(1-4/s)}} + (2\lambda)^3\left[(2\lambda)^{2-8/s} + 3\right] + 32\lambda^8\left[(2\lambda)^{2-8/s} + 3\right]^2 + \frac{1}{4}(2\lambda)^{8/s}\right],$$

which is at most $(2\lambda)^{2(k-2)(1-4/s)}$ since each term in the brackets on the right can be upper bounded by $1/4$ (using $\delta \le \frac{1}{64}(2\lambda)^{4t}$ and $\lambda \le 1/6$). □

6 Bounding the Statistics

Our use of the wide replacement walk introduces several complications on top of those encountered in Section 2 for the "vanilla" random walk. For this reason, we begin with a section where we collect the key ideas used in the proof of Claim 7. This will avoid repetitions and generally will facilitate a smoother reading of the proof, which is central to our main result.

6.1 The Tool Kit

Setup and Notations. We continue using the notation of Section 5.3. So specifically, A is a λ^2-expander and B is a $\lambda-$expander for a small parameter $\lambda > 0$. We have a fixed level $2(s+t+1)$ good pseudodistribution on $\{0,1\}^A$, and we defined vectors $\{\hat{\mathbf{v}}_S\}$ and $\{\mathbf{v}_S\}$ such that $\langle \hat{\mathbf{v}}_S, \hat{\mathbf{v}}_{S'}\rangle = \mathsf{Cov}(\mathsf{Y}_S, \mathsf{Y}_{S'})$, and $\langle \mathbf{v}_S, \mathbf{v}_{S'}\rangle = \tilde{\mathbb{E}}[\mathsf{Y}_S\mathsf{Y}_{S'}]$, for $S, S' \subset A$ such that $|S|, |S'| \leq s+t+1$. We will use the idenity

$$\tilde{\mathbb{E}}[\mathsf{Y}_S\mathsf{Y}_{S'}] = \langle \hat{\mathbf{v}}_S, \hat{\mathbf{v}}_{S'}\rangle + \tilde{\mathbb{E}}[\mathsf{Y}_S]\tilde{\mathbb{E}}[\mathsf{Y}_{S'}],$$

which follows from the definition of covariance. We will assume that $\left|\mathbb{E}_a\left[\tilde{\mathbb{E}}[\mathsf{Y}_a]\right]\right| \leq \lambda$ and additionally that the pseudodistribution is good, which means that for all $\ell, \ell' \in [s]$ and $k, k' \in [t]$,

$$\mathbb{E}_{\substack{\vec{a}\sim\mathsf{RW}_A^\ell, \vec{a}'\sim\mathsf{RW}_A^{\ell'} \\ \vec{a}''\sim\mathsf{WRW}_s^k(a_\ell) \\ \vec{a}'''\sim\mathsf{WRW}_s^{k'}(a'_{\ell'})}} \left[\left|\langle \hat{\mathbf{v}}_{\vec{a}\oplus\vec{a}''}, \hat{\mathbf{v}}_{\vec{a}'\oplus\vec{a}'''}\rangle\right| \leq \delta, \tag{6.1}$$

for a small parameter $\delta < \frac{1}{64}(2\lambda)^{4t}$. Some of the tools which we develop in this section will be applied several times during he proof of Claim 7 to different vector combinations. In order to state our tools with sufficient generality, throughout this section, we define vector-valued functions $\mathbf{z}_k, \mathbf{z}'_k$ on $A \times B$ for $\ell \in \mathbb{N}$ via

$$\mathbf{z}_k(a,b) := \mathbb{E}_{\substack{\vec{a}\sim\mathsf{WRW}_s^k(a,b) \\ S\sim\mathcal{D}(a_k)}}[\mathbf{v}_{\vec{a}\oplus S}]; \text{ and } \mathbf{z}'_k(a,b) := \mathbb{E}_{\substack{\vec{a}\sim\mathsf{WRW}_s^k(a,b) \\ S\sim\mathcal{D}'(a_k)}}[\mathbf{v}_{\vec{a}\oplus S}],$$

for distributions $\mathcal{D}$ and $\mathcal{D}'$ which, given $a \in A$, output subsets of A of size at most $t+1$, and where a_k is the final vertex in $\vec{a}$. Note the vector-valued functions $\mathbf{w}_k$ have this form with $\mathcal{D}$ the constant distribution which always outputs the emptyset. Define the statistics $\mu_{\mathbf{z}_k} := \left|\mathbb{E}_{a,b}\left[\mathbf{z}_k(a,b)\right]\right|$, and $\tau_{\mathbf{z}_k}$ such that $\mu^2_{\mathbf{z}_k} + \tau^2_{\mathbf{z}_k} = \mathbb{E}_{a,b}\left[|\mathbf{z}_k(a,b)|^2\right]$. Additionally, for $a \in A$, let $\mathbf{z}_k(a) := \mathbb{E}_b\left[\mathbf{z}_k(a,b)\right]$ and $\tau_{\mathbf{z}_k}(a)$ be so $|\mathbf{z}_k(a)|^2 + \tau_{\mathbf{z}_k}(a)^2 = \mathbb{E}_b\left[|\mathbf{z}_k(a,b)|^2\right]$. Note $\left|\mathbb{E}_a[\mathbf{z}_k(a)]\right| = \mu_{\mathbf{z}_k}$. Finally, let $\hat{\tau}_{\mathbf{z}_k}$ be such that $\mu^2_{\mathbf{z}_k} + \hat{\tau}^2_{\mathbf{z}_k} = \mathbb{E}_a\left[|\mathbf{z}_k(a)|^2\right]$. Define $\mu_{\mathbf{z}'_k}$, $\tau_{\mathbf{z}'_k}$, $\mathbf{z}'_k(a)$, $\tau_{\mathbf{z}'_k}(a)$, and $\hat{\tau}_{\mathbf{z}'_k}$ similarly.

Applying the EML Directly on Pseudoexpectations. For $k \in \mathbb{N}$, let $\mathbf{z}_k$ and $\mathbf{z}'_k$ be the vector-valued functions on $A \times B$ described above. Suppose we want to bound either

$$\mathsf{Q}_1 := \left|\mathbb{E}_{\substack{a\sim A, b\sim b' \\ \vec{a}\sim\mathsf{WRW}_s^k(a,b), S\sim\mathcal{D}(a_k) \\ \vec{a}'\sim\mathsf{WRW}_s^\ell(a,b'), S'\sim\mathcal{D}(a'_\ell)}} \left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}\oplus S}\mathsf{Y}_{\vec{a}'\oplus S'}]\right]\right|; \text{ or } \mathsf{Q}_2 := \left|\mathbb{E}_{\substack{a\sim a' \\ \vec{a}\sim\mathsf{WRW}_s^k(a), S\sim\mathcal{D}(a_k) \\ \vec{a}'\sim\mathsf{WRW}_s^\ell(a), S'\sim\mathcal{D}(a'_\ell)}} \left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}\oplus S}\mathsf{Y}_{\vec{a}'\oplus S'}]\right]\right|.$$

Since $\mathsf{Q}_1 = \left|\mathbb{E}_{a,b\sim b'}\left[\langle \mathbf{z}_k(a,b), \mathbf{z}'_\ell(a,b')\rangle\right]\right|$ and $\mathsf{Q}_2 = \left|\mathbb{E}_{a\sim a'}\left[\langle \mathbf{z}_k(a), \mathbf{z}'_\ell(a')\rangle\right]\right|$, these quantities are subject to bounds via the EML. In order to streamline the analysis in the next section, we will often apply the EML directly on the quantities in pseudoexpectation form; this will save considerable space because we will avoid having to convert between pseudoexpectation form and inner product form. Specifically, during the proof of Claim 7, we will frequently use:

$$\mathsf{Q}_1 \leq \left|\mathbb{E}_{\substack{a\sim A, b, b'\sim B \\ \vec{a}\sim\mathsf{WRW}_s^k(a,b), S\sim\mathcal{D}(a_k) \\ \vec{a}'\sim\mathsf{WRW}_s^\ell(a,b'), S'\sim\mathcal{D}(a'_\ell)}} \left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}\oplus S}\mathsf{Y}_{\vec{a}'\oplus S'}]\right]\right| + \lambda\tau_{\mathbf{z}_k}\tau_{\mathbf{z}'_\ell}; \tag{6.2}$$

and

$$\mathsf{Q}_2 \leq \left|\mathbb{E}_{\substack{a, a'\sim A \\ \vec{a}\sim\mathsf{WRW}_s^k(a), S\sim\mathcal{D}(a_k) \\ \vec{a}'\sim\mathsf{WRW}_s^\ell(a), S'\sim\mathcal{D}(a'_\ell)}} \left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}\oplus S}\mathsf{Y}_{\vec{a}'\oplus S'}]\right]\right| + \lambda^2\tau_{\mathbf{z}_k}\tau_{\mathbf{z}'_\ell}, \tag{6.2}$$

which follow from the EML and $\mathbb{E}_a\big[\tau_{\mathbf{z}_k}(a)\tau_{\mathbf{z}'_\ell}(a)\big], \hat{\tau}_{\mathbf{z}_k}\hat{\tau}_{\mathbf{z}'_\ell} \leq \tau_{\mathbf{z}_k}\tau_{\mathbf{z}'_\ell}$. Indeed, $\hat{\tau}_{\mathbf{z}_k} \leq \tau_{\mathbf{z}_k}$ holds because of Jensen's inequality:

$$\mu^2_{\mathbf{z}_k} + \hat{\tau}^2_{\mathbf{z}_k} = \mathbb{E}_a\big[|\mathbf{z}_k(a)|^2\big] \leq \mathbb{E}_{a,b}\big[|\mathbf{z}_k(a,b)|^2\big] = \mu^2_{\mathbf{z}_k} + \tau^2_{\mathbf{z}_k}.$$

Additionally, $\mathbb{E}_a\big[\tau_{\mathbf{z}_k}(a)\tau_{\mathbf{z}'_\ell}(a)\big]^2 \leq \mathbb{E}_a\big[\tau_{\mathbf{z}_k}(a)^2\big]\mathbb{E}_a\big[\tau_{\mathbf{z}'_\ell}(a)^2\big] \leq \tau^2_{\mathbf{z}_k}\tau^2_{\mathbf{z}'_\ell}$, where the first inequality is Cauchy-Schwarz and the second holds because

$$\mathbb{E}_a\big[|\mathbf{z}_k(a)|^2 + \tau_{\mathbf{z}_k}(a)^2\big] = \mathbb{E}_{a,b}\big[|\mathbf{z}_k(a,b)|^2\big] = \mu^2_{\mathbf{z}_k} + \tau^2_{\mathbf{z}_k},$$

and $\mu^2_{\mathbf{z}_k} \leq \mathbb{E}_a\big[|\mathbf{z}_k(a)|^2\big]$, again by Jensen's inequality.

Starting the Wide Replacement Walk in the Middle. Recall that for $(a,b) \in A \times B$, the distribution $\mathsf{WRW}^k_s(a,b)$ outputs $(a_0,\ldots,a_k) \in \mathsf{RW}^{k+1}_A$, generated as follows:

- a shifted random walk $(b_1,\ldots,b_k) \sim \widetilde{\mathsf{RW}}^k_B(b)$ is drawn;
- set $a_0 = a$ and for $i = 1,\ldots,k-1$, $a_{i+1} = \phi(a_i, \hat{b}_i)$.

Recall ϕ is the rotation map of A and $\hat{b}_i$ is the first coordinate of b_i when realized as an element of $[d]^s$. Because of the regularity of the explicit expander graphs used, it is possible to specify a distribution which is identical to WRW^k_s but which "starts the walk in the middle" by first drawing some $a_i \sim A$ for $i > 0$ and $b_i \sim b_{i+1}$ and then proceeding outward from a with wide replacement walks in both directions. Specifically, we will use that for all $k > s$:

$$\left\{ \vec{a} : \begin{array}{c} a \sim A, b \sim B \\ (a_0,\ldots,a_k) \sim \mathsf{WRW}^k_s(a,b) \end{array} \right\} \equiv \left\{ \vec{a} : \begin{array}{c} a \sim A, b \sim b' \\ (a_s, a_{s-1},\ldots,a_0) \sim \overleftarrow{\mathsf{WRW}}^s_s(a,b) \\ (a_s, a_{s+1},\ldots,a_k) \sim \mathsf{WRW}^{k-s}_s(a,b') \end{array} \right\}, \tag{6.3}$$

where both distributions output $\vec{a} = (a_0,\ldots,a_k)$, and where $\overleftarrow{\mathsf{WRW}}^s_s(a,b)$ is the same distribution as $\mathsf{WRW}^s_s(a,b)$ except that in the first step a reverse shifted random walk in B is drawn, rather than a shifted random walk as usual. We use the convention that $\overleftarrow{\mathsf{WRW}}^s_s(a,b)$ outputs $(a_s,\ldots,a_0)$ with $a_s = a$.

Pseudorandomness. In Claim 5 in Section 3, it is proven that for all $k \leq s$, the distribution which draws a $k-$length shifted random walk in B, $(b_1,\ldots,b_k)$, and outputs $(\hat{b}_1,\ldots,\hat{b}_k) \in [d]^k$ is identical to the uniform distribution on $[d]^k$. The same holds of course if a $k-$length reverse shifted random walk in B is drawn and $(\hat{b}_1,\ldots,\hat{b}_k)$ is output. It follows from this that short wide replacement walks (of length $\leq s$) are true random walks in A. This property was referred to as *pseudorandomness* in [Ta-17] and will be very useful for us. Specifically, we will use that for all $k \leq s$ and all $a \in A$,

$$\mathsf{WRW}^k_s(a) \equiv \mathsf{RW}^{k+1}_A(a) \equiv \overleftarrow{\mathsf{WRW}}^k_s(a). \tag{6.4}$$

Pseudodistribution Consistency. For all $T \subset S \subset A$ such that $|S| \leq 2(s+t+1)$, the distribution which draws $\sigma \sim \mathcal{O}(S)$ and outputs $\sigma|_T \in \{0,1\}^T$ is identical to $\mathcal{O}(T)$. It follows that for any distribution on subsets (T,S) of size at most $2(s+t+1)$ such that $T \subset S$,

$$\mathbb{E}_{T,S}\big[\tilde{\mathbb{E}}[\mathsf{Y}_T]\big] = \mathbb{E}_T\big[\tilde{\mathbb{E}}[\mathsf{Y}_T]\big], \tag{6.5}$$

where the second expectation is over T drawn according to the marginal of the distribution in the first expectation; the first pseudoexpectation is over $\mathcal{O}(S)$, the second is over $\mathcal{O}(T)$.

The "Ignore First Step" Trick. The observation here is that any two $\vec{a}, \vec{a}' \in \mathsf{WRW}_s^k(a,b)$ with the same starting pair (a,b) begin at the same vertex *and* take the same first step since both walk to $\phi(a,\hat{b})$. This allows for a simplificaion to the expectations which show up in the standard deviation bounds. Specifically, let $\mathbf{z}_k$ be the vector-valued function on $A \times B$ introduced above. The key identity is:

$$\tau^2_{\mathbf{z}_k} \leq \mathbb{E}_{\substack{a\sim A \\ \vec{a},\vec{a}'\sim \mathsf{WRW}_s^{k-1}(a) \\ S\sim\mathcal{D}(a_{k-1}), S'\sim\mathcal{D}(a'_{k-1})}} \left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}\oplus S}\mathsf{Y}_{\vec{a}'\oplus S'}]\right] + \lambda^2\tau^2_{\mathbf{z}_{k-1}}. \tag{6.6}$$

This is proved as follows

$$\begin{aligned}
\tau^2_{\mathbf{z}_k} &\leq \mu^2_{\mathbf{z}_k} + \tau^2_{\mathbf{z}_k} = \mathbb{E}_{a,b}\left[|\mathbf{z}_k(a,b)|^2\right] = \mathbb{E}_{\substack{a\sim A, b\sim B \\ \vec{a}\sim\mathsf{WRW}_s^k(a,b), S\sim\mathcal{D}(a_k) \\ \vec{a}'\sim\mathsf{WRW}_s^k(a,b), S'\sim\mathcal{D}(a'_k)}} \left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}\oplus S}\mathsf{Y}_{\vec{a}'\oplus S'}]\right] \\
&= \mathbb{E}_{\substack{a\sim A, b\sim B \\ \vec{a}\sim\mathsf{WRW}_s^k(a,b), S\sim\mathcal{D}(a_k) \\ \vec{a}'\sim\mathsf{WRW}_s^k(a,b), S'\sim\mathcal{D}(a'_k)}} \left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}_{1:k}\oplus S}\mathsf{Y}_{\vec{a}'_{1:k}\oplus S'}]\right] \overset{(6.5)}{=} \mathbb{E}_{\substack{a\sim A, b\sim_{B^2} b' \\ \vec{a}\sim\mathsf{WRW}_s^{k-1}(a,b) \\ \vec{a}'\sim\mathsf{WRW}_s^{k-1}(a,b') \\ S\sim\mathcal{D}(a_{k-1}), S'\sim\mathcal{D}(a'_{k-1})}} \left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}\oplus S}\mathsf{Y}_{\vec{a}'\oplus S'}]\right] \\
&\overset{(6.2)}{\leq} \mathbb{E}_{\substack{a\sim A \\ \vec{a},\vec{a}'\sim\mathsf{WRW}_s^{k-1}(a) \\ S\sim\mathcal{D}(a_{k-1}), S'\sim\mathcal{D}(a'_{k-1})}} \left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}\oplus S}\mathsf{Y}_{\vec{a}'\oplus S'}]\right] + \lambda^2\tau^2_{\mathbf{z}_{k-1}}.
\end{aligned}$$

The first equality on the second line holds because $\mathsf{Y}_{\vec{a}\oplus S}\mathsf{Y}_{\vec{a}'\oplus S'} = \mathsf{Y}_{\vec{a}_{1:k}\oplus S}\mathsf{Y}_{\vec{a}'_{1:k}\oplus S'}$ since $\alpha_0 = \alpha'_0$ holds with probability 1 over $\mathcal{O}(\vec{a}\oplus S\oplus\vec{a}'\oplus S')$ (since $a_0 = a'_0$). The second equality on the second line holds by pseudodistribution consistency, since if we do not care about the common starting point of two $k-$step wide replacement walks which take the same first step, then we can instead draw their common second vertex randomly, and two random neighbors (in B) of the original b and take $(k-1)-$step wide replacement walks.

Generalizations of Lemma 1. Consider the vector-valued function $\mathbf{z}_k$ on $A\times B$ introduced above, and recall that for $a\in A$, $\mathbf{z}_k(a) = \mathbb{E}_b[\mathbf{z}_k(a,b)]$. Note that when $k\leq s$, we have

$$\mathbf{z}_k(a) = \mathbb{E}_{\substack{b\sim B, \vec{a}\sim\mathsf{WRW}_s^k(a,b) \\ S\sim\mathcal{D}(a_k)}}[\mathbf{v}_{\vec{a}\oplus S}] \overset{(6.4)}{=} \mathbb{E}_{\substack{\vec{a}\sim\mathsf{RW}_A^{k+1}(a) \\ S\sim\mathcal{D}(a_k)}}[\mathbf{v}_{\vec{a}\oplus S}].$$

Define and recall the statistics $\varepsilon_{\mathbf{z}_k}$, $\mu_{\mathbf{z}_k}$, and $\hat{\tau}_{\mathbf{z}_k}$ so that

$$\varepsilon_{\mathbf{z}_k} := \left|\mathbb{E}_{\substack{\vec{a}\sim\mathsf{RW}_A^{k+1} \\ S\sim\mathcal{D}(a_{k+1})}}\left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}\oplus S}]\right]\right|; \text{ and } \mu^2_{\mathbf{z}_k} + \hat{\tau}^2_{\mathbf{z}_k} = \mathbb{E}_{\substack{a\sim A \\ \vec{a},\vec{a}'\sim\mathsf{RW}_A^{k+1}(a) \\ S\sim\mathcal{D}(a_{k+1}), S'\sim\mathcal{D}(a'_{k+1})}}\left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}\oplus S}\mathsf{Y}_{\vec{a}'\oplus S'}]\right].$$

Then as long as $\mathcal{D}$ is such that

$$\mathbb{E}_{\substack{a\sim A, \vec{a}\sim\mathsf{RW}_A^k \\ S\sim\mathcal{D}(a_k)}}\left[|\langle\hat{\mathbf{v}}_a, \hat{\mathbf{v}}_{\vec{a}\oplus S}\rangle|\right] \leq \delta; \text{ and } \mathbb{E}_{\substack{\vec{a},\vec{a}'\sim\mathsf{RW}_A^k \\ S\sim\mathcal{D}(a_k), S'\sim\mathcal{D}(a'_k)}}\left[|\langle\hat{\mathbf{v}}_{\vec{a}\oplus S}, \hat{\mathbf{v}}_{\vec{a}'\oplus S'}\rangle|\right] \leq \delta,$$

then we have that for all $k\geq 1$,

$$\varepsilon_{\mathbf{z}_k} \leq \sqrt{\delta} + \frac{1}{2}(2\lambda)^k(\varepsilon_{\mathbf{z}_0} + \lambda\hat{\tau}_{\mathbf{z}_0}); \text{ and } \mu^2_{\mathbf{z}_k} + \hat{\tau}^2_{\mathbf{z}_k} \leq 4\delta + (2\lambda)^{2(k-1)}(\varepsilon_{\mathbf{z}_0} + \lambda\hat{\tau}_{\mathbf{z}_0})^2. \tag{6.7}$$

These bounds are proved using the same methods as used in the proof of Lemma 1. Specifically, we use an EML calculation to prove that for all $k \geq 1$,

$$\varepsilon_{\mathbf{z}_k} \leq \delta + \lambda\varepsilon_{\mathbf{z}_{k-1}} + \lambda^2\hat{\tau}_{\mathbf{z}_{k-1}};\ \text{and}\ \mu^2_{\mathbf{z}_k} + \hat{\tau}^2_{\mathbf{z}_k} \leq \delta + \varepsilon^2_{\mathbf{z}_{k-1}} + \lambda^4\hat{\tau}^2_{\mathbf{z}_{k-1}}. \tag{$*$}$$

From $(*)$, an inductive argument can be used to show that for all $k \geq 1$,

$$\varepsilon_{\mathbf{z}_k} \leq \sqrt{\delta} + \frac{1}{2}(2\lambda)^k(\varepsilon_{\mathbf{z}_0} + \lambda\hat{\tau}_{\mathbf{z}_0});\ \text{and}\ \hat{\tau}_{\mathbf{z}_k} \leq 3\sqrt{\delta} + (2\lambda)^{k-1}(\varepsilon_{\mathbf{z}_0} + \lambda\hat{\tau}_{\mathbf{z}_0}). \tag{$**$}$$

This establishes the bound on $\varepsilon_{\mathbf{z}_k}$ stated in (6.7); the bound on $\mu^2_{\mathbf{z}_k} + \hat{\tau}^2_{\mathbf{z}_k}$ in (6.7) is recovered by plugging in the bounds of $(**)$ into the right hand side of the bound for $\mu^2_{\mathbf{z}_k} + \hat{\tau}^2_{\mathbf{z}_k}$ given in $(*)$ and simplifying. Deriving $(**)$ from $(*)$ is a simple induction once note that $(*)$ implies $\hat{\tau}_{\mathbf{z}_k} \leq \sqrt{\delta} + \varepsilon_{\mathbf{z}_{k-1}} + \lambda^2\hat{\tau}_{\mathbf{z}_{k-1}}$, since $\hat{\tau}_{\mathbf{z}_k} \leq (\mu^2_{\mathbf{z}_k} + \hat{\tau}^2_{\mathbf{z}_k})^{1/2}$. Plugging in $k = 1$ gives the base case of $(**)$; the induction step is established by plugging in the induction hypothesis into the right hand sides of $(*)$ and simplifying. Finally, to derive the bounds in $(*)$ we note that for $k \geq 1$ we have

$$\begin{aligned}
\varepsilon_{\mathbf{z}_k} &= \left|\mathbb{E}_{\substack{a\sim a' \\ \vec{a}\sim\mathsf{RW}^k_A(a') \\ S\sim\mathcal{D}(a_k)}}\left[\tilde{\mathbb{E}}[\mathsf{Y}_a\mathsf{Y}_{\vec{a}\oplus S}]\right]\right| \overset{\text{(EML)}}{\leq} \left|\mathbb{E}_{\substack{a\sim A,\vec{a}\sim\mathsf{RW}^k_A \\ S\sim\mathcal{D}(a_k)}}\left[\tilde{\mathbb{E}}[\mathsf{Y}_a\mathsf{Y}_{\vec{a}\oplus S}]\right]\right| + \lambda^2\hat{\tau}_{\mathbf{z}_{k-1}} \\
&\leq \delta + \lambda\varepsilon_{\mathbf{z}_{k-1}} + \lambda^2\hat{\tau}_{\mathbf{z}_{k-1}} \\
\mu^2_{\mathbf{z}_k} + \hat{\tau}^2_{\mathbf{z}_k} &= \mathbb{E}_{\substack{a\sim A \\ \vec{a},\vec{a}'\sim\mathsf{RW}^{k+1}_A(a) \\ S\sim\mathcal{D}(a_{k+1}),S'\sim\mathcal{D}(a'_{k+1})}}\left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}\oplus S}\mathsf{Y}_{\vec{a}'\oplus S'}]\right] \overset{(6.5)}{=} \mathbb{E}_{\substack{a\sim_{A^2}a' \\ \vec{a}\sim\mathsf{RW}^k_A(a),S\sim\mathcal{D}(a_k) \\ \vec{a}'\sim\mathsf{RW}^k_A(a'),S'\sim\mathcal{D}(a'_k)}}\left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}\oplus S}\mathsf{Y}_{\vec{a}'\oplus S'}]\right] \\
&\overset{\text{(EML)}}{\leq} \mathbb{E}_{\substack{\vec{a}\sim\mathsf{RW}^k_A,S\sim\mathcal{D}(a_k) \\ \vec{a}'\sim\mathsf{RW}^k_A,S'\sim\mathcal{D}(a'_k)}}\left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}\oplus S}\mathsf{Y}_{\vec{a}'\oplus S'}]\right] + \lambda^4\hat{\tau}^2_{\mathbf{z}_{k-1}} \leq \delta + \varepsilon^2_{\mathbf{z}_{k-1}} + \lambda^4\hat{\tau}^2_{\mathbf{z}_{k-1}}.
\end{aligned}$$

6.2 Proof of Claim 7

We establish the four bounds of Claim 7 separately. In each part we will define a parametrized family of "helper functions" $\mathbf{z}_k$, for $k \in \mathbb{N}$. These are vector-valued functions on $A \times B$, which will always have the form

$$\mathbf{z}_k(a,b) := \mathbb{E}_{\substack{\vec{a}\sim\mathsf{WRW}^k_s(a,b) \\ S\sim\mathcal{D}(a_k)}}\left[\mathbf{v}_{\vec{a}\oplus S}\right],$$

for some distribution $\mathcal{D}$ which, given input $a \in A$, outputs a subset $S \subset A$ of size at most t. As usual, we define the statistics $\mu_{\mathbf{z}_k} = \left|\mathbb{E}_{a,b}[\mathbf{z}_k(a,b)]\right|$ and $\tau_{\mathbf{z}_k}$ such that $\mu^2_{\mathbf{z}_k} + \tau^2_{\mathbf{z}_k} = \mathbb{E}_{a,b}\left[|\mathbf{z}_k(a,b)|^2\right]$. Additionally, for $a \in A$, let $\mathbf{z}_k(a) := \mathbb{E}_b[\mathbf{z}_k(a,b)]$, and let $\hat{\tau}_{\mathbf{z}_k}$ be such that $\mu^2_{\mathbf{z}_k} + \hat{\tau}^2_{\mathbf{z}_k} = \mathbb{E}_a\left[|\mathbf{z}_k(a)|^2\right]$. Finally, let $\varepsilon_{\mathbf{z}_k} := \left|\mathbb{E}_{\substack{\vec{a}\sim\mathsf{WRW}^k_s \\ S\sim\mathcal{D}(a_k)}}\left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}\oplus S}]\right]\right|$. We stress that the definition of $\mathbf{z}_k$ will vary in the four parts.

Part 1 – Bounding ε_k when $k \leq s$. Let $\mathbf{z}_k(a,b) := \mathbb{E}_{\vec{a}\sim\mathsf{WRW}^k_s(a,b)}[\mathbf{v}_{\vec{a}}]$ (so for all $a \in A$, the distribution $\mathcal{D}(a)$ outputs the emptyset with probability 1). Note that when $k \leq s$,

$$\mathbf{z}_k(a) = \mathbb{E}_{\substack{b\sim B \\ \vec{a}\sim\mathsf{WRW}^k_s(a,b)}}\left[\mathbf{v}_{\vec{a}}\right] \overset{(6.4)}{=} \mathbb{E}_{\vec{a}\sim\mathsf{RW}^{k+1}(a)}[\mathbf{v}_{\vec{a}}],$$

and so has the form required to apply generalizations of Lemma 1. Note, $\varepsilon_{\mathbf{z}_0} \leq \lambda$ by assumption, and $\hat{\tau}_{\mathbf{z}_0} \leq 1$ is trivial. Thus,

$$\varepsilon_k = \varepsilon_{\mathbf{z}_k} \overset{(6.7)}{\leq} \sqrt{\delta} + \frac{1}{2}(2\lambda)^{k+1}.$$

Part 2 – Bounding τ_k when $k \le s$. Continuing with $\mathbf{z}_k(a,b) := \mathbb{E}_{\vec{a}\sim\mathsf{WRW}_s^k(a,b)}[\mathbf{v}_{\vec{a}}]$, we have

$$\begin{aligned}\tau_k^2 &\overset{(6.6)}{\le} \mathbb{E}_{\substack{a\sim A\\ \vec{a},\vec{a}'\sim\mathsf{WRW}_s^{k-1}(a)}}\big[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}\mathsf{Y}_{\vec{a}'}]\big] + \lambda^2\tau_{k-1}^2 \overset{(6.4)}{=} \mathbb{E}_{\substack{a\sim A\\ \vec{a},\vec{a}'\sim\mathsf{RW}_A^k(a)}}\big[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}\mathsf{Y}_{\vec{a}'}]\big] + \lambda^2\tau_{k-1}^2.\\ &= \mu^2_{\mathbf{z}_{k-1}} + \hat{\tau}^2_{\mathbf{z}_{k-1}} + \lambda^2\tau_{k-1}^2 \overset{(6.7)}{\le} 4\delta + (2\lambda)^{2(k-1)} + \lambda^2\tau_{k-1}^2,\end{aligned}$$

from which it follows (*e.g.*, by induction) that $\tau_k \le 3\sqrt{\delta} + 2(2\lambda)^{k-1}$.

Part 3 – Bounding ε_k when $k > s$. Let us keep $\mathbf{z}_k(a,b) := \mathbb{E}_{\vec{a}\sim\mathsf{WRW}^k(a,b)}[\mathbf{v}_{\vec{a}}]$ as before and additionally define the function $\mathbf{z}'_k(a,b) := \mathbb{E}_{\vec{a}\sim\overleftarrow{\mathsf{WRW}}_s^s(a,b)}[\mathbf{v}_{\vec{a}_{0:s-1}}]$. Note that

$$\begin{aligned}\tau^2_{\mathbf{z}'_s} &\le \mathbb{E}_{\substack{a\sim A, b\sim B\\ \vec{a},\vec{a}'\sim\overleftarrow{\mathsf{WRW}}_s^s(a,b)}}\big[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}_{0:s-1}}\mathsf{Y}_{\vec{a}'_{0:s-1}}]\big] \overset{(6.5)}{=} \mathbb{E}_{\substack{a\sim A, b\sim_{B^2}b'\\ \vec{a}\sim\overleftarrow{\mathsf{WRW}}_s^{s-1}(a,b)\\ \vec{a}'\sim\overleftarrow{\mathsf{WRW}}_s^{s-1}(a,b')}}\big[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}\mathsf{Y}_{\vec{a}'}]\big]\\ &\overset{(6.2)+(6.4)}{\le} \mathbb{E}_{\substack{a\sim A\\ \vec{a},\vec{a}'\sim\mathsf{RW}_A^s(a)}}\big[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}\mathsf{Y}_{\vec{a}'}]\big] + \lambda^2\tau_{k-1}^2 \overset{(6.5)}{=} \mathbb{E}_{\substack{a\sim_{A^2}a'\\ \vec{a}\sim\mathsf{RW}_A^{s-1}(a)\\ \vec{a}'\sim\mathsf{RW}_A^{s-1}(a')}}\big[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}\mathsf{Y}_{\vec{a}'}]\big] + \lambda^2\tau_{k-1}^2\\ &\overset{(6.2)+(6.1)}{\le} \delta + \varepsilon_{s-2}^2 + \lambda^4\tau_{s-2}^2 + \lambda^2\tau_{s-1}^2,\end{aligned}$$

and so $\tau_{\mathbf{z}'_s} \le \sqrt{\delta} + \varepsilon_{s-2} + \lambda^2\tau_{s-2} + \lambda\tau_{s-1} \le 3\sqrt{\delta} + 2(2\lambda)^{s-1}$, via the bounds in parts 1 and 2. We can now bound ε_k as follows:

$$\begin{aligned}\varepsilon_k &= \big|\mathbb{E}_{\vec{a}\sim\mathsf{WRW}_s^k}\big[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}]\big]\big| \overset{(6.3)}{=} \left|\mathbb{E}_{\substack{a\sim A, b\sim b'\\ \vec{a}\sim\overleftarrow{\mathsf{WRW}}_s^s(a,b)\\ \vec{a}'\sim\mathsf{WRW}_s^{k-s}(a,b')}}\Big[\tilde{\mathbb{E}}\big[\mathsf{Y}_{\vec{a}_{0:s-1}}\mathsf{Y}_{\vec{a}'}\big]\Big]\right|\\ &\overset{(6.2)+(6.4)+(6.5)}{\le} \left|\mathbb{E}_{\substack{a\sim a'\\ \vec{a}\sim\mathsf{RW}_A^s(a)\\ \vec{a}'\sim\mathsf{WRW}_s^{k-s}(a')}}\big[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}\mathsf{Y}_{\vec{a}'}]\big]\right| + \lambda\tau_{\mathbf{z}'_s}\tau_{k-s}\\ &\overset{(6.2)+(6.1)}{\le} \delta + \varepsilon_{s-1}\varepsilon_{k-s} + \lambda^2\tau_{s-1}\tau_{k-s} + \lambda\tau_{\mathbf{z}'_s}\tau_{k-s} \le \sqrt{\delta} + \frac{1}{2}(2\lambda)^s\big[\varepsilon_{k-s} + 3\tau_{k-s}\big].\end{aligned}$$

Part 4 – Bounding τ_k when $k > s$. For $k > s$, define the quantity η_k so that

$$\eta_k^2 := \mathbb{E}_{\substack{a\sim A\\ \vec{a},\vec{a}'\sim\mathsf{WRW}_s^k(a)}}\big[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}\mathsf{Y}_{\vec{a}'}]\big].$$

Since $\tau_k^2 \le \eta_{k-1}^2 + \lambda^2\tau_{k-1}^2$ by (6.6), it suffices to bound η_{k-1}^2. For this purpose, for $k, \ell \in \mathbb{N}$, define

$$\mathbf{z}_{\ell,k}(a,b) := \mathbb{E}_{\substack{\vec{a}\sim\overleftarrow{\mathsf{WRW}}_s^\ell(a,b)\\ \vec{a}'\sim\mathsf{WRW}_s^{k-1}(a_0)}}\big[\mathbf{v}_{\vec{a}\oplus\vec{a}'}\big];\ \text{and}\ \mathbf{z}'_{\ell,k}(a,b) := \mathbb{E}_{\substack{\vec{a}\sim\overleftarrow{\mathsf{WRW}}_s^\ell(a,b)\\ \vec{a}'\sim\mathsf{WRW}_s^{k-1}(a_\ell)}}\big[\mathbf{v}_{\vec{a}_{0:\ell-1}\oplus\vec{a}'}\big].$$

Since $k > s$ will remain fixed for us, we write $\mathbf{z}_\ell(a,b)$ and $\mathbf{z}'_\ell(a,b)$ instead of $\mathbf{z}_{\ell,k}(a,b)$ and $\mathbf{z}'_{\ell,k}(a,b)$. Note that when $\ell \le s$,

$$\mathbf{z}_\ell(a) = \mathbb{E}_{\substack{b\sim B,\vec{a}\sim\overleftarrow{\mathsf{WRW}}_s^\ell(a,b)\\ \vec{a}'\sim\mathsf{WRW}^{k-1}(a_0)}}\big[\mathbf{v}_{\vec{a}\oplus\vec{a}'}\big] \overset{(6.4)}{=} \mathbb{E}_{\substack{\vec{a}\sim\mathsf{RW}_s^{\ell+1}(a)\\ \vec{a}'\sim\mathsf{WRW}^{k-1}(a_{\ell+1})}}\big[\mathbf{v}_{\vec{a}\oplus\vec{a}'}\big],$$

and so has the correct form needed to apply the generalization of Lemma 1. Note $\varepsilon_{\mathbf{z}_0} = \varepsilon_{k-2}$ and $\tau^2_{\mathbf{z}_0} = \hat{\tau}^2_{\mathbf{z}_0} \le \eta^2_{k-1}$, thus $\varepsilon_{\mathbf{z}_0} + \lambda\hat{\tau}_{\mathbf{z}_0} \le \varepsilon_{k-2} + \lambda\eta_{k-1}$. The same argument used to derive the bound in (6.6) shows that for $\ell \le s$, $\tau^2_{\mathbf{z}_\ell}, \tau^2_{\mathbf{z}'_\ell} \le \mu^2_{\mathbf{z}_{\ell-1}} + \hat{\tau}^2_{\mathbf{z}_{\ell-1}} + \lambda^2\tau^2_{\mathbf{z}_{\ell-1}} \le 4\delta + (2\lambda)^{2(\ell-2)}(\varepsilon_{k-2} + \lambda\eta_{k-1})^2 + \lambda^2\tau^2_{\mathbf{z}_{\ell-1}}$, using also (6.7). This implies (*e.g.*, by induction) that for all $\ell \le s$, $\tau_{\mathbf{z}_\ell}, \tau_{\mathbf{z}'_\ell} \le 3\sqrt{\delta} + 2(2\lambda)^{\ell-2}[\varepsilon_{k-2} + \lambda\eta_{k-1}]$. We can now bound η^2_{k-1} as follows:

$$\begin{aligned}
\eta^2_{k-1} &= \mathbb{E}_{\substack{a\sim A \\ \vec{a},\vec{a}'\sim \mathsf{WRW}_s^{k-1}(a)}} \left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}}\mathsf{Y}_{\vec{a}'}]\right] \overset{(6.3)}{=} \mathbb{E}_{\substack{a\sim A, b\sim b' \\ \vec{a}\sim \overleftarrow{\mathsf{WRW}}_s^s(a,b) \\ \vec{a}'\sim \mathsf{WRW}_s^{k-1}(a_0) \\ \vec{a}''\sim \mathsf{WRW}_s^{k-s}(a,b')}} \left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}_{0:s-1}\oplus\vec{a}'}\mathsf{Y}_{\vec{a}''}]\right] \\
&\overset{(6.2)+(6.4)+(6.5)}{\le} \mathbb{E}_{\substack{a\sim a' \\ \vec{a}\sim \mathsf{RW}_A^s(a) \\ \vec{a}'\sim \mathsf{WRW}_s^{k-1}(a_s) \\ \vec{a}''\sim \mathsf{WRW}_s^{k-s}(a')}} \left[\tilde{\mathbb{E}}[\mathsf{Y}_{\vec{a}\oplus\vec{a}'}\mathsf{Y}_{\vec{a}''}]\right] + \lambda\tau_{\mathbf{z}'_s}\tau_{k-s} \\
&\overset{(6.2)+(6.1)+(6.7)}{\le} \sqrt{\delta} + \frac{1}{2}(2\lambda)^{s-1}(\varepsilon_{k-s} + 3\tau_{k-s})(\varepsilon_{k-2} + \lambda\eta_{k-1}).
\end{aligned}$$

This implies (via completing the square)

$$\eta^2_{k-1} \le 2\sqrt{\delta} + (2\lambda)^{s-1}\left[\varepsilon_{k-s} + 3\tau_{k-s}\right]\varepsilon_{k-2} + \frac{\lambda^2}{2}(2\lambda)^{2(s-1)}\left[\varepsilon_{k-s} + 3\tau_{k-s}\right]^2,$$

as desired.

Acknowledgements

We thank Luca Trevisan for several illuminating conversations during the early stages of this work, and Alon Rosen for support and encouragement.

References

[ABN+92] Noga Alon, Jehoshua Bruck, Joseph Naor, Moni Naor, and Ron M. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Trans. Inf. Theory*, 38(2):509–516, 1992.

[AGHP92] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple construction of almost k-wise independent random variables. *Random Struct. Algorithms*, 3(3):289–304, 1992.

[AJQ+20] Vedat Levi Alev, Fernando Granha Jeronimo, Dylan Quintana, Shashank Srivastava, and Madhur Tulsiani. List decoding of direct sum codes. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1412–1425. SIAM, 2020.

[AKK+08] Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K. Vishnoi. Unique games on expanding constraint graphs are easy: extended abstract. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 21–28. ACM, 2008.

[Alo21] Noga Alon. Explicit expanders of every degree and size. *Comb.*, 41(4):447–463, 2021.

[Bog12] Andrej Bogdanov. A different way to improve the bias via expanders. In *Topics in (and out of) the theory of computing, Lecture, 2012*, 2012.

[BRS11] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 472–481. IEEE Computer Society, 2011.

[BT11] Avraham Ben-Aroya and Amnon Ta-Shma. A combinatorial construction of almost-ramanujan graphs using the zig-zag product. *SIAM J. Comput.*, 40(2):267–290, 2011.

[Gil52] E. N. Gilbert. A comparison of signalling alphabets. *The Bell System Technical Journal*, 31(3):504–522, 1952.

[GR08] Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Trans. Inf. Theory*, 54(1):135–150, 2008.

[JQST20] Fernando Granha Jeronimo, Dylan Quintana, Shashank Srivastava, and Madhur Tulsiani. Unique decoding of explicit ε-balanced codes near the gilbert-varshamov bound. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 434–445. IEEE, 2020.

[JST21] Fernando Granha Jeronimo, Shashank Srivastava, and Madhur Tulsiani. Near-linear time decoding of ta-shma's codes via splittable regularity. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1527–1536. ACM, 2021.

[Jus72] J. Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18(5):652–656, 1972.

[Plo60] Morris Plotkin. Binary codes with specified minimum distance. *IRE Transactions on Information Theory*, 6(4):445–450, 1960.

[RR22] Silas Richelson and Sourya Roy. Analyzing ta-shma's code via the expander mixing lemma. *CoRR*, abs/2201.11166, 2022.

[Ta-17] Amnon Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 238–251. ACM, 2017.

[Var57] R. R. Varshamov. Estimate of the number of signals in error correcting codes. *Docklady Akad. Nauk, S.S.S.R.*, 117:739–741, 1957.

A Omitted Proofs

We include the proofs of Claims 1 and 6.

Claim 1 (Restated). *Suppose* X *and* Y *are jointly distributed real-valued random variables, and moreover that* X *is supported on a set of size* 2*. Then* $\mathsf{Cov}(\mathsf{X},\mathsf{Y})^2/\mathsf{Var}(\mathsf{X}) = \mathsf{Var}(\mathsf{Y}) - \mathbb{E}_{x\sim\mathsf{X}}\big[\mathsf{Var}(\mathsf{Y}|x)\big]$.

Proof. Without loss of generality, we can assume that $\mathbb{E}[\mathsf{X}] = \mathbb{E}[\mathsf{Y}] = 0$ and $\mathbb{E}[\mathsf{X}^2] = 1$. With these simplifications, we must show that $\mathbb{E}[\mathsf{XY}]^2 = \mathbb{E}_{x\sim\mathsf{X}}\big[\mathbb{E}[\mathsf{Y}|x]^2\big]$. Note that as X has $\mathbb{E}[\mathsf{X}] = 0$, $\mathbb{E}[\mathsf{X}^2] = 1$ and has support of size 2, there exists $a > 0$ such that $\mathsf{X} = a$ with probability $1/(a^2+1)$ and $\mathsf{X} = -1/a$ with probability $a^2/(a^2+1)$. Let $A := \mathbb{E}[\mathsf{Y}|\mathsf{X} = a]$ and $B := \mathbb{E}[\mathsf{Y}|\mathsf{X} = -1/a]$. This gives us $\mathbb{E}[\mathsf{XY}]^2 = \frac{a^2}{(a^2+1)^2}\cdot(A-B)^2$, and $\mathbb{E}_{x\sim\mathsf{X}}\big[\mathbb{E}[\mathsf{Y}|x]^2\big] = \frac{1}{a^2+1}\cdot(A^2+a^2B^2)$. A calculation shows that these are equal if and only if $A + a^2B = 0$, which holds because $A + a^2B = (a^2+1)\cdot\mathbb{E}[\mathsf{Y}]$. □

Claim 6 (Restated). *If* $\mathcal{O}$ *is the pseudodistribution obtained during Step 1 of* $\mathcal{D}_k(\tilde{\mathbf{y}})$*, then for all* $\mathbf{x} \in \mathsf{LIST}(\tilde{\mathbf{y}}, \rho)$*,*

$$\mathbb{E}_{\vec{a},\vec{a}'\sim\mathsf{WRW}_s^t}\big[\tilde{\mathbb{E}}[(-1)^{\mathbf{x}_{\vec{a}}\oplus\vec{\alpha}\oplus\mathbf{x}_{\vec{a}'}\oplus\vec{\alpha}'}]\big] \geq \frac{1}{4}\rho^4.$$

Proof. For a pseudodistribution $\mathcal{O}$ on $\{0,1\}^A$, $\tilde{\mathbf{y}} \in \{0,1\}^{\mathsf{WRW}_s^t}$ and $\mathbf{x} \in \{0,1\}^A$, define

$$\mathsf{val}(\mathcal{O}) := \mathbb{E}_{\vec{a},\vec{a}'\sim\mathsf{WRW}_s^t}\big[\tilde{\mathbb{E}}[(-1)^{\vec{\alpha}\oplus\vec{\alpha}'}]^2\big];\ \mathsf{val}(\mathcal{O},\mathbf{x}) := \mathbb{E}_{\vec{a},\vec{a}'\sim\mathsf{WRW}_s^t}\big[\tilde{\mathbb{E}}[(-1)^{\mathbf{x}_{\vec{a}}\oplus\vec{\alpha}\oplus\mathbf{x}_{\vec{a}'}\oplus\vec{\alpha}'}]\big],$$

where both of the pseudoexpectations are over $(\vec{\alpha},\vec{\alpha}') \sim \mathcal{O}(\vec{a}\cup\vec{a}')$. Now, let $\mathcal{O}$ be the pseudodistribution on $\{0,1\}^A$ which is recovered by the SDP solver during Step 1 of the execution of $\mathcal{D}_k(\tilde{\mathbf{y}})$ for $\mathbf{y} \in \{0,1\}^{\mathsf{WRW}_s^t}$, and let $\mathbf{x} \in \mathsf{LIST}(\tilde{\mathbf{y}},\rho)$. Consider the pseudodistribution $\hat{\mathcal{O}}_{\mathbf{x}}$ which is a convex combination of $\mathcal{O}$ and the constant pseudodistribution which answers according to $\mathbf{x}$. So specifically, on input $S \subset A$, $\hat{\mathcal{O}}_{\mathbf{x}}(S)$ returns a sample from $\mathcal{O}(S)$ with probability γ, and returns $\mathbf{x}_S$ with probability $1-\gamma$, for some $\gamma \in (0,1)$ which we will fix later. A straightforward computation shows that $\mathsf{val}(\hat{\mathcal{O}}_{\mathbf{x}}) = \gamma^2\mathsf{val}(\mathcal{O}) + 2\gamma(1-\gamma)\mathsf{val}(\mathcal{O},\mathbf{x}) + (1-\gamma)^2$, so since $\mathcal{O}$ is an optimal pseudodistribution for minimizing $\mathsf{val}(\mathcal{O})$ subject to the constraints holding, it must be that $\mathsf{val}(\mathcal{O}) - \frac{1}{16}\rho^8 \leq \mathsf{val}(\hat{\mathcal{O}}_{\mathbf{x}})$ (it is straightforward to check that the constraints also hold for $\hat{\mathcal{O}}_{\mathbf{x}}$, since $\mathbf{x} \in \mathsf{LIST}(\tilde{\mathbf{y}},\rho)$). Setting $\gamma = 1 - \frac{1}{4}\rho^4$ and rearranging implies

$$\frac{1}{2}\mathsf{val}(\mathcal{O}) - \frac{1}{4}\rho^4 \leq \mathsf{val}(\mathcal{O},\mathbf{x}).$$

Finally, we show that $\mathsf{val}(\mathcal{O}) \geq \rho^4$, which implies that $\mathsf{val}(\mathcal{O},\mathbf{x}) \geq \frac{1}{4}\rho^4$, completing the proof. This final bound holds because $\mathcal{O}$ satisfies the constraints of the SDP. Indeed,

$$\begin{aligned}
\rho^4 &\leq \mathbb{E}_{\vec{a}\sim\mathsf{WRW}_s^t}\big[\tilde{\mathbb{E}}[(-1)^{\tilde{\mathbf{y}}_{\vec{a}}\oplus\vec{\alpha}}]\big]^4 \\
&\leq \Big(\mathbb{E}_{\vec{a},\vec{a}'\sim\mathsf{WRW}_s^t}\big[\mathsf{Cov}\big((-1)^{\tilde{\mathbf{y}}_{\vec{a}}\oplus\vec{\alpha}},(-1)^{\tilde{\mathbf{y}}_{\vec{a}'}\oplus\vec{\alpha}'}\big)\big] + \mathbb{E}_{\vec{a}\sim\mathsf{WRW}_s^t}\big[\tilde{\mathbb{E}}[(-1)^{\tilde{\mathbf{y}}_{\vec{a}}\oplus\vec{\alpha}}]\big]^2\Big)^2 \\
&= \mathbb{E}_{\vec{a},\vec{a}'\sim\mathsf{WRW}_s^t}\big[\tilde{\mathbb{E}}[(-1)^{\tilde{\mathbf{y}}_{\vec{a}}\oplus\vec{\alpha}\oplus\tilde{\mathbf{y}}_{\vec{a}'}\oplus\vec{\alpha}'}]\big]^2 \leq \mathbb{E}_{\vec{a},\vec{a}'\sim\mathsf{WRW}_s^t}\big[\tilde{\mathbb{E}}[(-1)^{\vec{\alpha}\oplus\vec{\alpha}'}]^2\big] = \mathsf{val}(\mathcal{O}),
\end{aligned}$$

where the final inequality is Jensen's inequality, and the inequality on the second line holds because $\mathbb{E}_{\vec{a},\vec{a}'\sim\mathsf{WRW}_s^t}\big[\mathsf{Cov}\big((-1)^{\tilde{\mathbf{y}}_{\vec{a}}\oplus\vec{\alpha}},(-1)^{\tilde{\mathbf{y}}_{\vec{a}'}\oplus\vec{\alpha}'}\big)\big] = \mathsf{Var}\big(\mathbb{E}_{\vec{a}\sim\mathsf{WRW}_s^t}\big[\tilde{\mathbb{E}}[(-1)^{\tilde{\mathbf{y}}_{\vec{a}}\oplus\vec{\alpha}}]\big]\big) > 0$. □

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il