Electronic Colloquium on Computational Complexity, Report No. 31 (2025)

Error-Correction of Matrix Multiplication Algorithms

Shuichi Hirahara
National Institute of Informatics
s_hirahara@nii.ac.jp

Nobutaka Shimizu
Institute of Science Tokyo
shimizu.n.ah@m.titech.ac.jp

Abstract

Given an efficient algorithm that correctly computes a *tiny* fraction of the entries of the matrix multiplication of a *small* fraction of two matrices, can one design an efficient algorithm that computes matrix multiplication *exactly* for *all* the matrices? In this paper, we present such "worst-case exact to average-case approximate" reductions that transform any algorithm that correctly computes a tiny fraction of the entries of the multiplication of two uniformly random matrices over a finite field into a randomized worst-case algorithm that computes matrix multiplication for all the matrices. Under non-uniform reductions, we present an optimal reduction that error-corrects an algorithm whose output has expected Hamming distance $1 - \frac{1}{p} - \varepsilon$ to the multiplication of two random matrices over a finite field of size p for any positive constant $\varepsilon > 0$. Under uniform reductions, we present efficient reductions that correct a $(1 - \varepsilon)$-fraction of errors over a field of size p for all $\varepsilon > 0$ and for all sufficiently large p. We also present an optimal uniform reduction for the Online Matrix-Vector Multiplication problem.

The non-uniform reduction is based on a new and simple proof of Yao's XOR lemma for multi-output functions, whose complexity overhead is independent of the length of the output.

ISSN 1433-8092

Contents

1 Introduction

Matrix multiplication is one of the most fundamental algebraic primitives that lies at the center of scientific computation. The computational complexity of matrix multiplication has been actively studied from both theoretical and practical perspectives. After a long line of research that started from the work of Strassen [Str69], the current best algorithm computes the multiplication of two $n \times n$ matrices in time $O(n^\omega)$, where $2 \leq \omega \leq 2.3716$ (see [DWZ23; VXXZ24; ADWXXZ25] and references therein), which is much faster than the naïve algorithm that runs in time $O(n^3)$. The algorithms developed in this line of research are asymptotically fast, but are not efficient in practice (where n is reasonably small) because they suffer from a huge leading constant hidden in the big O notation. Another line of research, started from the "Four-Russians" algorithm, has developed combinatorial algorithms for Boolean Matrix Multiplication, which tend to be more efficient in practice (see, e.g., [BW12; AFKLM24] and references therein).

From a practical perspective, we do not have to confine ourselves to the standard computing hardware. Specialized hardware devices, such as GPUs, enable multiplying large matrices efficiently in practice [VD08]. There have been proposals for architectures to compute matrix multiplication based on physical phenomena, such as optical devices [ZDCDHSZGQCRZ22], thermodynamics [CADMGASCMS23], and divisible materials (e.g., water) [Val24]. Such computational devices would significantly suffer from a high level of noise and errors. It is therefore natural and important to ask the following question.

> Given a black-box device that *approximately* computes matrix multiplication, can one design an efficient algorithm that computes matrix multiplication *exactly*?

The first result in this direction was recently presented by Gola, Shinkar, and Singh [GSS24]. Assuming that there is a black-box algorithm $\mathcal{O}$ that, given as input uniformly random matrices $A, B \sim \mathbb{F}^{n\times n}$ over a finite field $\mathbb{F}$, outputs a matrix C such that a $\frac{8}{9}$-fraction of the entries of C agree with $A \cdot B$ in expectation, they designed a randomized algorithm $M^{\mathcal{O}}$ that computes the matrix multiplication of all the matrices. Their reduction is based on a simple modification of the worst-case to average-case reduction for matrix multiplication developed by Blum, Luby, and Rubinfeld [BLR93], which cannot tolerate a large fraction of errors. Error-tolerant worst-case to average-case reductions for matrix multiplication have been recently developed by Asadi, Golovnev, Gur, and Shinkar [AGGS22] and Hirahara and Shimizu [HS23]. These reductions do not correct the type of errors such that some entries are incorrectly answered; the reductions assume average-case algorithms to compute *all* the $n \times n$ entries of $A \cdot B$ correctly for a small fraction of $n \times n$ matrices A and B. It is an outstanding problem whether one can obtain a worst-case algorithm from an average-case algorithm that computes a *tiny* fraction of the $n \times n$ entries of $A \cdot B$ correctly for a small fraction of $n \times n$ matrices A and B.

1.1 Our Results

In this paper, we fully resolve this problem for non-uniform algorithms. For two matrices A and $B \in \mathbb{F}_p^{n\times n}$ over a finite field of size p, let $\mathrm{dist}(A, B)$ denote the fraction of the entries $(i, j) \in \{1, \cdots n\}^2$ such that the (i, j)-th entries of A and B disagree. The trivial algorithm $\mathcal{O}$ that outputs

the all-0 matrix achieves the expected distance[1]

$$\mathop{\mathbb{E}}_{A,B\sim\mathbb{F}_p^{n\times n}}[\mathrm{dist}(\mathcal{O}(A,B),AB)] \le 1-\frac{1}{p},$$

where the expectation is taken over uniform random matrices A and B over $\mathbb{F}_p^{n\times n}$. Any algorithm that achieves an expected distance smaller by an additive constant $\varepsilon > 0$ can be transformed into a worst-case *non-uniform* algorithm (i.e., an algorithm that takes an advice string for each input length, or almost equivalently, a Boolean circuit).

Theorem 1.1. *Let $\mathbb{F}_p$ be a field of prime order p and $\varepsilon > 0$ be a constant. Suppose that there is an oracle $\mathcal{O}$ such that for all sufficiently large $n \in \mathbb{N}$,*

$$\mathop{\mathbb{E}}_{A,B\sim\mathbb{F}_p^{n\times n}}[\mathrm{dist}(\mathcal{O}(A,B),AB)] \le 1-\frac{1}{p}-\varepsilon,$$

where A and B are uniformly chosen from $\mathbb{F}_p^{n\times n}$. Then, for all sufficiently large $n \in \mathbb{N}$, there exists a randomized $O(\log n)$-query oracle circuit C of size $\widetilde{O}(n^2)$ such that for every $A, B \in \mathbb{F}_p^{n\times n}$,

$$\Pr_C\left[C^{\mathcal{O}}(A,B) = AB\right] \ge \frac{2}{3},$$

where the probability is over the internal randomness of C. Moreover, there exists a randomized polynomial-time algorithm that, on input 1^n, prints the description of such a circuit C with probability $1-o(1)$.

Here, the $\widetilde{O}(\cdot)$ notation hides a $\mathsf{polylog}(n)$ factor. Note that the constant $\frac{2}{3}$ can be amplified to $1-o(1)$ by the standard technique of repetition and a majority vote.

In words, this theorem shows that given a black-box computational device $\mathcal{O}$ that "approximates" matrix multiplication, one can obtain a nearly-linear-time algorithm for matrix multiplication with *preprocessing*: In the preprocessing phase, one can compute the description of C in polynomial time. Then, such a description C can be used to compute the matrix multiplication of *every* pair of matrices (A, B) in nearly linear time. The hidden polylogarithmic factor in $\widetilde{O}(n^2)$ and the hidden constant in $O(\log n)$ are not very large (see Theorem 3.1 for details), and thus we expect that Theorem 1.1 could be useful in practice.

As an immediate corollary, we obtain the following equivalence.

Corollary 1.2. *Let $\mathbb{F}_p$ be a field of prime order p and $\varepsilon > 0$ be a constant. There exists a randomized circuit C of size $\widetilde{O}(n^2)$ such that*

$$\mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ C}}[\mathrm{dist}(C(A,B),AB)] \le 1-\frac{1}{p}-\varepsilon$$

if and only if there exists a randomized circuit C of size $\widetilde{O}(n^2)$ such that for every $A, B \in \mathbb{F}_p^{n\times n}$,

$$\Pr_C[C(A,B) = AB] \ge 1-o(1).$$

[1] We mention in passing that the expected distance $1-\frac{1}{n}$ can also be achieved trivially by computing the first row of AB in time $O(n^2)$.

Next, we present uniform reductions. When a field size p is sufficiently large, we obtain a highly efficient *uniform* reduction.

Theorem 1.3. *Let $\mathbb{F}_p$ be a finite field of order $p = p(n)$ and $\varepsilon = \varepsilon(n) > 0$ be a parameter such that $p > n/\varepsilon^2$. Suppose there exists an oracle $\mathcal{O}$ such that for all sufficiently large $n \in \mathbb{N}$,*

$$\mathop{\mathbb{E}}_{A,B\sim\mathbb{F}_p^{n\times n}}[\mathrm{dist}(\mathcal{O}(A,B), AB)] \le 1-\varepsilon.$$

Then, there exists a randomized $\mathsf{poly}(1/\varepsilon)\cdot O(\log n)$-query oracle algorithm $M^{\mathcal{O}}$ that runs in time $\mathsf{poly}(1/\varepsilon)\cdot\mathsf{polylog}(p)\cdot\widetilde{O}(n^2)$ and satisfies, for all large $n\in\mathbb{N}$ and for any $A,B\in\mathbb{F}_p^{n\times n}$,

$$\Pr_{M^{\mathcal{O}}}\left[M^{\mathcal{O}}(A,B) = AB\right] \ge \frac{2}{3}.$$

This theorem is superior to Theorem 1.1 in that the reduction is uniform, and that the number of queries is independent of p, whereas that of Theorem 1.1 (Theorem 3.1) depends on p. The only deficiency is that the field size p must be large. When p is small, we obtain another uniform reduction that corrects errors almost optimally up to a factor of 2.

Theorem 1.4. *Let $\mathbb{F}_p$ be a finite field of prime order $p = p(n)$ and $\varepsilon = \varepsilon(n) > 0$ be a parameter. Suppose that there exists an oracle $\mathcal{O}$ such that for all sufficiently large $n \in \mathbb{N}$,*

$$\mathop{\mathbb{E}}_{A,B\sim\mathbb{F}_p^{n\times n}}[\mathrm{dist}(\mathcal{O}(A,B), AB)] \le 1-\frac{2}{p}-\varepsilon,$$

where A and B are uniformly chosen from $\mathbb{F}_p^{n\times n}$. Then, there exists a randomized $2^{\mathsf{poly}(p,1/\varepsilon)}\cdot O(\log n)$-query randomized $2^{p^{\mathsf{poly}(p,1/\varepsilon)}}\cdot\widetilde{O}(n^2)$-time oracle algorithm M such that for all sufficiently large $n\in\mathbb{N}$, and for every $A,B\in\mathbb{F}_p^{n\times n}$,

$$\Pr_{M^{\mathcal{O}}}\left[M^{\mathcal{O}}(A,B) = AB\right] \ge \frac{2}{3}.$$

1.2 Online Matrix-Vector Multiplication

We also present an optimal uniform reduction for the online matrix-vector multiplication (OMv) problem [HKNS15]. The OMv problem asks to preprocess an $n\times n$ matrix A in polynomial time in the preprocessing phase, and then, in the query phase, to compute $A\cdot v$ efficiently for n-dimensional vectors v. We consider an average-case and $\mathbb{F}_p$ variant of OMv, where the arithmetic operations are over a finite field $\mathbb{F}_p$ [HLS22; AGGS22]. Any algorithm that achieves an expected distance better than $1-\frac{1}{p}$ can be transformed into a worst-case algorithm.

Theorem 1.5. *Let $\mathbb{F}_p$ be a field of prime order p and let $\varepsilon > 0$ be a constant. Then, the following are equivalent.*

- *There exists a randomized data-structure algorithm M that runs in time $\widetilde{O}(n)$ in the query phase such that for all sufficiently large $n\in\mathbb{N}$,*

$$\mathop{\mathbb{E}}_{\substack{A\sim\mathbb{F}_p^{n\times n}\\ v\sim\mathbb{F}_p^n\\ M}}[\mathrm{dist}(M(A;v),\ Av)] \le 1-\frac{1}{p}-\varepsilon.$$

- *There exists a randomized data-structure algorithm M' that runs in time $\widetilde{O}(n)$ in the query phase such that for all sufficiently large $n \in \mathbb{N}$ and all $A \in \mathbb{F}_p^{n \times n}$ and $v \in \mathbb{F}_p^n$,*

$$\Pr_{M'}\big[M'(A; v) = Av\big] \geq \frac{2}{3}.$$

2 Techniques

Our reductions are based on two "disjoint" sets of techniques. One is based on the theory of hardness amplification developed in, e.g., [IJK09; IJKW10; HS22; HS23; HS24]. This yields the non-uniform reduction of Theorem 1.1, for which we present an overview in Section 2.1. The other is based on error-correcting codes and error-tolerant worst-case to average-case reductions of [HS23]. This yields uniform reductions, for which we present an overview in Sections 2.2 and 2.3.

Notation. For $n \in \mathbb{N}$, let $[n]$ denote $\{1, \ldots, n\}$. For a matrix $A \in \mathbb{F}_p^{n \times n}$, we denote by $A_{i,j}$ the (i, j)-th entry of A. For two vectors $a = (a_1, \cdots, a_n)$ and $b = (b_1, \cdots, b_n) \in \mathbb{F}_p^n$, let $\mathrm{dist}(a, b) = \Pr_{i \sim [n]}[a_i \neq b_i]$ denote the normalized Hamming distance. We say that a is r-close to b if $\mathrm{dist}(a, b) \leq r$. For a finite set S, we write $x \sim S$ to denote that x is chosen uniformly at random from S.

2.1 The Non-Uniform Reduction

The best error-tolerant worst-case to average-case reduction for matrix multiplication of [HS23], which improves [AGGS22], is based on the techniques from hardness amplification. Prototypical theorems in hardness amplification are Yao's XOR lemma and the direct product theorem (see [GNW11] for a survey). Yao's XOR lemma states that if a function $f\colon \{0,1\}^n \to \{0,1\}$ is *mildly average-case hard* (meaning that every small circuit fails to compute f on a δ-fraction of inputs for a small $\delta > 0$), then the K-wise XOR function $f^{\oplus K}\colon (\{0,1\}^n)^K \to \{0,1\}$ is *strongly average-case hard* (meaning that every small circuit fails to compute $f^{\oplus K}$ on a $(1/2 - \varepsilon)$-fraction of inputs for a small $\varepsilon > 0$), where $f^{\oplus K}(x_1, \cdots, x_K) \coloneqq f(x_1) \oplus \cdots \oplus f(x_K)$ for $(x_1, \cdots, x_K) \in (\{0,1\}^n)^K$. Similarly, the direct product theorem states that for a mildly average-case hard function $f\colon \{0,1\}^n \to \{0,1\}$, the K-wise direct product $f^K\colon (\{0,1\}^n)^K \to \{0,1\}^K$ is strongly average-case hard (meaning that every small circuit fails to compute f^K on a $(1 - \varepsilon)$-fraction of inputs for a small $\varepsilon > 0$), where $f^K(x_1, \ldots, x_K) \coloneqq (f(x_1), \ldots, f(x_K))$ for $(x_1, \cdots, x_K) \in (\{0,1\}^n)^K$. In a recent line of research [HS22; HS23; HS24], the techniques of hardness amplification have been used to show "hardness self-amplification" for natural problems, such as matrix multiplication, triangle counting, and the planted clique problem. Given this line of research, it is natural to apply a similar technique to our settings, which, however, seemed highly non-trivial as noted in [GSS24].[2]

Our key insight is that the K-wise XOR structure is hidden in matrix multiplication. Consider the case of $p = 2$, and let $N = nK$. For two matrices $\bar{A}$ and $\bar{B} \in \mathbb{F}_2^{N \times N}$, partition these matrices so that

$$\bar{A} = \begin{bmatrix} A_1 & \cdots & A_K \end{bmatrix} \in \left(\mathbb{F}_2^{N \times n}\right)^{1 \times K}, \quad \bar{B} = \begin{bmatrix} B_1 \\ \vdots \\ B_K \end{bmatrix} \in \left(\mathbb{F}_2^{n \times N}\right)^{K \times 1}.$$

[2] The authors of [GSS24] wrote "we do not see how to apply the Direct-Product theorem to our setting of the problem." The key innovation here is that we use Yao's XOR lemma instead of the direct product theorem used in [HS23].

Then, we have

$$\bar{A} \cdot \bar{B} = \sum_{k \in [K]} A_k \cdot B_k = (A_1 \cdot B_1) \oplus \cdots \oplus (A_K \cdot B_K) = f(A_1, B_1) \oplus \cdots \oplus f(A_K, B_K),$$

where, in the last equality, we defined $f \colon \mathbb{F}_2^{N \times n} \times \mathbb{F}_2^{n \times N} \to \mathbb{F}_2^{N \times N}$ to be the function such that $f(A, B) = A \cdot B$. This means that the matrix multiplication of $\bar{A}$ and $\bar{B}$ coincides with the output of the K-wise XOR function $f^{\oplus K}$ on input $((A_1, B_1), \ldots, (A_K, B_K))$. By Yao's XOR lemma, we may expect that the mild average-case hardness of f implies the strong average-case hardness of $f^{\oplus K}$; taking its contrapositive, an average-case algorithm that computes $f^{\oplus K}$ on a small fraction of inputs can be transformed into an average-case algorithm that computes f on a large fraction of inputs.[3]

Unfortunately, all the known proofs of Yao's XOR lemma (e.g., [GNW11]) are applicable only to *single-output* functions $f \colon \{0,1\}^n \to \{0,1\}$. Here, we present a new, simple, and generalized proof of Yao's XOR lemma for *multi-output* functions. For a multi-output function $f \colon \{0,1\}^n \to \{0,1\}^m$, let $f^{\oplus k}$ denote the function such that $f^{\oplus k}(x_1, \ldots, x_k) = f(x_1) \oplus \cdots \oplus f(x_k)$, where $\oplus$ denotes the bitwise XOR. For a string $y \in \{0,1\}^m$ and $\ell \in [m]$, let y_ℓ denote the ℓ-th bit of y.

Theorem 2.1 (see also Theorem A.1)**.** *Let $f \colon \{0,1\}^n \to \{0,1\}^m$ be a function and $\mathcal{L}$ be a distribution over $[m]$. Suppose that there exists a circuit C' of size $s' \geq m$ such that for all $\ell \in [m]$,*

$$\Pr_{\bar{x} \sim (\{0,1\}^n)^k} \left[C'(\bar{x})_\ell = f^{\oplus k}(\bar{x})_\ell \right] \geq \frac{1}{2} + \varepsilon,$$

where $k \geq O\big(\log(1/\varepsilon)/(\varepsilon\delta)^2\big)$. Then, there exists a circuit C of size $s = s' \cdot O\big(\log(1/\delta)/\varepsilon^2\big)$ such that

$$\Pr_{\substack{x \sim \{0,1\}^n \\ \ell \sim \mathcal{L}}} [C(x)_\ell = f(x)_\ell] \geq 1 - \delta.$$

It is possible to prove a result analogous to Theorem 2.1 by applying Yao's XOR lemma to the ℓ-th bit of f for all $\ell \in [m]$ independently. However, such a proof would incur the $O(m)$ factor of overhead in circuit size s. This is not tolerable in reductions for matrix multiplication, where m corresponds to the size of matrices. The significance of Theorem 2.1 is that the circuit size s is at most $O(s')$ for constants ε and $\delta > 0$. Moreover, the proof of Theorem 2.1 is fairly simple and may have pedagogical value in the case of $m = 1$. At a high level, the circuit C is constructed by embedding an input $x \in \{0,1\}^n$ into one element of random tuples $\bar{x} \sim (\{0,1\}^n)^k$, evaluating C' on input $\bar{x}$, and taking a majority vote of the outputs of C' over independent random choices. The proof can be found in Section A.

There are two issues to apply Theorem 2.1 to matrix multiplication. The first issue is that we need to assume that for *all* entries $\ell = (i, j) \in [N]^2$, an oracle $\mathcal{O}$ computes the (i, j)-th entry of the multiplication $\bar{A} \cdot \bar{B}$ of random $N \times N$ matrices $(\bar{A}, \bar{B})$ with probability $\frac{1}{2} + \varepsilon$. This issue can be fixed by using the simple technique of random permutation developed by Gola, Shinkar, and Singh [GSS24] (see Lemma 3.2). The second issue is how to generalize it to a larger field size $p > 2$. The

[3] This reduces the matrix multiplication of *rectangular* matrices $(A, B) \in \mathbb{F}_2^{N \times n} \times \mathbb{F}_2^{n \times N}$ to the matrix multiplication of *square* matrices $(\bar{A}, \bar{B}) \in \mathbb{F}_2^{N \times N}$. By combining this with a trivial worst-case reduction from square matrix multiplication to rectangular matrix multiplication, we may obtain a worst-case to average-case reduction for the same input size; see the proof of Theorem 3.1 for details.

proof of Theorem 2.1 is crucially based on the majority vote, which cannot be generalized to the case where the error is larger than $\frac{1}{2}$. We overcome this difficulty, by presenting a simple reduction that "distributes" an error of an oracle $\mathcal{O}$ among $\mathbb{F}_p$ uniformly at random (see Lemma 3.4). Details can be found in Section 3.

2.2 Online Matrix-Vector Multiplication

Next, we present the ideas for constructing uniform reductions. For the sake of clarity, we first explain the ideas for online matrix-vector multiplication (see Section 5 for details), and then explain how the ideas can be modified to apply to matrix multiplication (see Section 4.2 for details).

2.2.1 Exact-to-Approximate Reduction via List-Decodable Code

We start with a "worst-case exact to worst-case approximate" reduction that transforms an algorithm that computes a vector that is $(1-\alpha)$-close to the multiplication Av of *every* matrix A and *every* vector v into an algorithm that computes OMv *exactly*. This reduction can be obtained by a simple application of a list-decodable error-correcting code. Consider a linear error-correcting code $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ that takes a message of length n and outputs its codeword. Since Enc is a linear map, it can be written in the form $\mathsf{Enc}\colon x \mapsto Qx$ for some matrix $Q \in \mathbb{F}_p^{N\times n}$. For an efficient reduction, we need an error-correcting code with $N = O_\alpha(n)$ that is computable and list-decodable within radius $1-\alpha$ in nearly linear time. That is, the function $x \mapsto \mathsf{Enc}(x)$ can be computed in $\widetilde{O}_{p,\alpha}(n)$ time, and for a given $\tilde{y} \in \mathbb{F}_p^N$, the number of vectors $x \in \mathbb{F}_p^n$ satisfying $\mathrm{dist}(\mathsf{Enc}(x), \tilde{y}) \le 1-\alpha$ is at most $O_{p,\alpha}(1)$, and all such x can be enumerated in time $\widetilde{O}_{p,\alpha}(N)$.

For a matrix $A \in \mathbb{F}_p^{n\times n}$ consisting of column vectors $a_1, \ldots, a_n \in \mathbb{F}_p^n$, the matrix $QA \in \mathbb{F}_p^{N\times n}$ consists of column vectors $\mathsf{Enc}(a_j)$ for $j = 1, \ldots, n$ and can be computed in time $\widetilde{O}(n^2)$. Using this, we obtain the following approximate-to-exact reduction for OMv:

1. Let M_{pre} be the preprocess algorithm that receives a matrix $A \in \mathbb{F}_p^{n\times n}$ and performs the preprocessing, and M_{query} be the algorithm that receives a vector $v \in \mathbb{F}_p^n$ and computes a vector that is $(1-\alpha)$-close to Av.

2. In the preprocessing of our reduction, for a given matrix $A \in \mathbb{F}_p^{n\times n}$ compute $QA \in \mathbb{F}_p^{N\times n}$ in $\widetilde{O}(n^2)$ time, and then run the preprocess M_{pre} on input QA.

3. For a given query $v \in \mathbb{F}_p^n$, compute $\tilde{w} = M_{\mathrm{query}}(v) \in \mathbb{F}_p^N$. Since this vector satisfies $\mathrm{dist}(\tilde{w}, QAv) = \mathrm{dist}(\tilde{w}, \mathsf{Enc}(Av)) \le 1-\alpha$, using the list-decoding algorithm of Enc, we obtain a list of $O_{p,\alpha}(1)$ vectors in $\mathbb{F}_p^n$ that contains Av in time $\widetilde{O}(N)$. Using an efficient verification algorithm for OMv, we can identify Av in this list in time $\widetilde{O}(N)$.

2.2.2 Worst-Case-to-Average-Case Reduction via Full-Rank Partition

We now upgrade the previous reduction to a "worst-case exact to average-case approximate" reduction. In the average-case version of OMv, the input (A, v) to an algorithm is chosen uniformly at random from $\mathbb{F}_p^{n\times n} \times \mathbb{F}_p^n$. Given an approximate average-case solver that outputs a vector $\tilde{w} \in \mathbb{F}_p^n$ such that

$$\mathop{\mathbb{E}}_{A,v}[\mathrm{dist}(\tilde{w}, Av)] \le 1-\alpha,$$

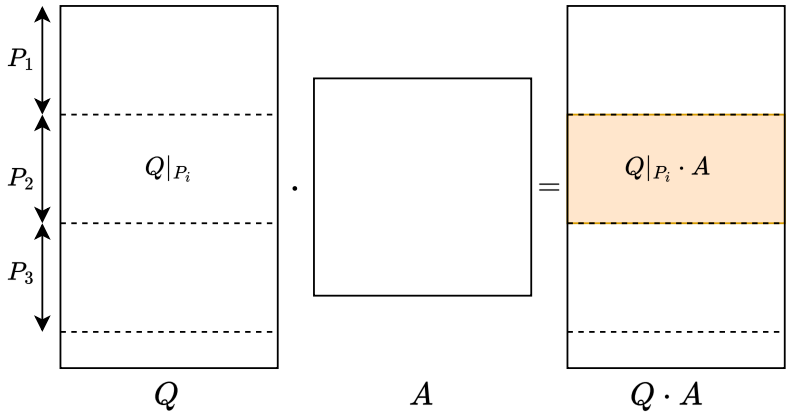

Figure 1: The marginal distribution of each submatrix $Q|_{P_i} \cdot A$ is uniform over $\mathbb{F}_p^{m \times n}$ if $A \sim \mathbb{F}_p^{n \times n}$ is a uniformly random matrix.

our goal is to construct a worst-case and exact algorithm for OMv.

The reason why the reduction of Section 2.2.1 is not a worst-case-to-average-case reduction is that $QA \in \mathbb{F}_p^{N \times n}$ (which is given to the average-case solver) is not uniformly distributed even if A is chosen uniformly at random. We thus need a way to ensure that queries to the average-case solver are uniformly distributed. To this end, we simply divide the set $[N]$ of the row indices of $QA \in \mathbb{F}_p^{N \times n}$ into several subsets so that the marginal distribution of the submatrix of QA induced by each subset is uniformly distributed for a uniformly random matrix A. More precisely, for an encoding $\mathsf{Enc}\colon \mathbb{F}_p^n \ni x \mapsto Qx \in \mathbb{F}_p^N$ and parameters $m \leq n$ and $\delta > 0$, an (m, δ)-*full-rank partition* $\mathcal{P} = P_1 \sqcup \cdots \sqcup P_a \sqcup P'$ is a partition of the set $[N]$ such that

- $|P_i| = m$ for each $i \in [a]$,
- $|P'| \leq \delta N$, and
- for each $i \in [a]$, the submatrix $Q|_{P_i} \in \mathbb{F}_p^{m \times n}$ consisting of all the row vectors of Q whose index belongs to P_i is full row rank.

The parameters m, δ are chosen so that $m = \Theta(n)$ and $\delta \ll \alpha$. In this overview, we assume $\delta = 0$ (i.e., $P' = \emptyset$) for simplicity. Suppose that the encoding Enc has an $(m, 0)$-full-rank partition $\mathcal{P} = P_1 \sqcup \cdots \sqcup P_a$. For a uniformly random matrix $A \sim \mathbb{F}_p^{n \times n}$, when each row of $QA \sim \mathbb{F}_p^{N \times n}$ is divided into submatrices along $\mathcal{P}$, for each $i \in [a]$, the submatrix $Q|_{P_i} A$ of QA is uniformly distributed over $\mathbb{F}_p^{m \times n}$ (see Fig. 1). This implies that, for each $i \in [a]$, the vector $\tilde{w}_i \in \mathbb{F}_p^m$ obtained by running the approximate average-case solver on input $(Q|_{P_i} A, v)$ satisfies $\mathbb{E}_{A,v}[\mathrm{dist}(\tilde{w}_i, Q|_{P_i} Av)] \leq 1 - \alpha$. By concatenating the vectors $\tilde{w}_i$ for all $i \in [a]$, we can obtain a vector $\tilde{w} \in \mathbb{F}^N$ such that $\mathbb{E}_{A,v}[\mathrm{dist}(\tilde{w}, QAv)] \leq 1 - \alpha$.

Fix an approximate average-case solver. We call an input (A, v) *good* if it satisfies $\mathrm{dist}(\tilde{w}, QAv) \leq 1 - \alpha/2$. By Markov's inequality, the probability that $(A, v) \sim \mathbb{F}_p^{n \times n} \times \mathbb{F}_p^n$ is good is at least $\alpha/2$. For any good (A, v), the vector $\tilde{w} \in \mathbb{F}_p^N$ obtained using the approximate average-case solver is $(1 - \alpha/2)$-close to QAv, and thus we can use the exact-to-approximation reduction given in Section 2.2.1 to compute Av in time $\widetilde{O}(N)$. In this way, we obtain an average-case algorithm that

computes OMv *exactly* for the $(\alpha/2)$-fraction of good inputs (A, v). Finally, applying the error-tolerant worst-case-to-average-case reduction for OMv of Hirahara and Shimizu [HS23], we obtain a worst-case exact algorithm for OMv.

2.3 Matrix Multiplication

Building on the ideas developed for OMv, we now explain how to construct uniform reductions for matrix multiplication.

2.3.1 Reduction for Matrix Multiplication via Left-Right Encoding

In the reduction for OMv, for an input $A \in \mathbb{F}_p^{n\times n}$, we considered the matrix $QA \in \mathbb{F}_p^{N\times n}$ obtained by applying a list-decodable error-correcting code $\mathsf{Enc}\colon \mathbb{F}_p^n \ni x \mapsto Qx \in \mathbb{F}_p^N$. The crucial point is that the vector $\mathsf{Enc}(Av) = QA \cdot v$ coincides with the encoding of the answer Av for a given vector $v \in \mathbb{F}_p^n$.

The same idea is applicable to matrix multiplication if some encoding $\mathsf{Enc}^\star(AB) \in \mathbb{F}_p^{N\times N}$ of the answer AB on input $A, B \in \mathbb{F}_p^{n\times n}$ can be expressed as the product of some pair of two matrices. We present a natural encoding scheme $\mathsf{Enc}^\star$ with such a property, which we call *left-right encoding*. [4] For an arbitrary error-correcting code $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ that can be written as $\mathsf{Enc}\colon x \mapsto Qx$ for a matrix $Q \in \mathbb{F}_p^{N\times n}$, we define the left-right encoding $\mathsf{Enc}^\star\colon \mathbb{F}_p^{n\times n} \to \mathbb{F}_p^{N\times N}$ as

$$\mathsf{Enc}^\star\colon A \mapsto QAQ^\top.$$

For matrices A and $B \in \mathbb{F}_p^{n\times n}$, the left-right encoding of the product AB is given by $\mathsf{Enc}^\star(AB) = QABQ^\top = (QA)\cdot(QB^\top)^\top$. Thus, by querying an oracle on input $\left(QA, (QB^\top)^\top\right)$, we may obtain an $N \times N$ matrix that is $(1-\alpha)$-close to $\mathsf{Enc}^\star(AB)$. Assuming that $\mathsf{Enc}^\star$ is list-decodable within radius $1-\alpha$, we can compute a list of matrices that contains AB. The correct solution can be identified using Freivalds' randomized verification algorithm (Lemma 4.26).

This argument is a natural extension of the reduction for OMv described in Section 2.2.1 to matrix multiplication. Combined with the full-rank partition with a slight modification of the arguments in Section 2.2.2, we obtain a worst-case-to-average-case reduction for matrix multiplication.

2.3.2 List-Decoding of Left-Right Encoding

In Section 2.3.1, we assumed that $\mathsf{Enc}^\star$ is list-decodable within radius $1-\alpha$. Gopalan, Guruswami, and Raghavendra [GGR11, Theorem 4.7] showed that this is indeed the case, assuming that the original encoding Enc is list-decodable within radius $1-\alpha/2$. Here, we provide a simple proof of this fact in our case of matrix multiplication. Specifically, we consider the problem of reconstructing the matrix $AB \in \mathbb{F}_p^{n\times n}$ given as input $(A, B) \in \left(\mathbb{F}_p^{n\times n}\right)^2$ and a matrix $\widetilde{C} \in \mathbb{F}_p^{N\times N}$ that is $(1-\alpha)$-close to $\mathsf{Enc}^\star(AB)$.

The reconstruction is outlined in Fig. 2. Let $A, B \in \mathbb{F}_p^{n\times n}, \widetilde{C} \in \mathbb{F}_p^{N\times N}$ be the input such that $\mathrm{dist}(\widetilde{C}, \mathsf{Enc}^\star(AB)) \le 1-\alpha$. Since the matrix $\widetilde{C}$ satisfies $\mathrm{dist}(\widetilde{C}, Q\cdot ABQ^\top) \le 1-\alpha$, from Markov's inequality, at least $(\alpha/2)\cdot N$ column vectors $\tilde{c}_j \in \mathbb{F}_p^N$ of $\widetilde{C}$ satisfy $\mathrm{dist}(\tilde{c}_j, \mathsf{Enc}(d_j))) \le 1-\alpha/2$, where $d_j \in \mathbb{F}_p^n$ is the j-th column vector of $ABQ^\top \in \mathbb{F}_p^{n\times N}$. Let us call such column *good*. By using the list-decoding algorithm of Enc, we can compute the set of vectors that contains d_j for every good

[4] In the literature of error-correcting code, the left-right encoding is called a tensor product code [GGR11].

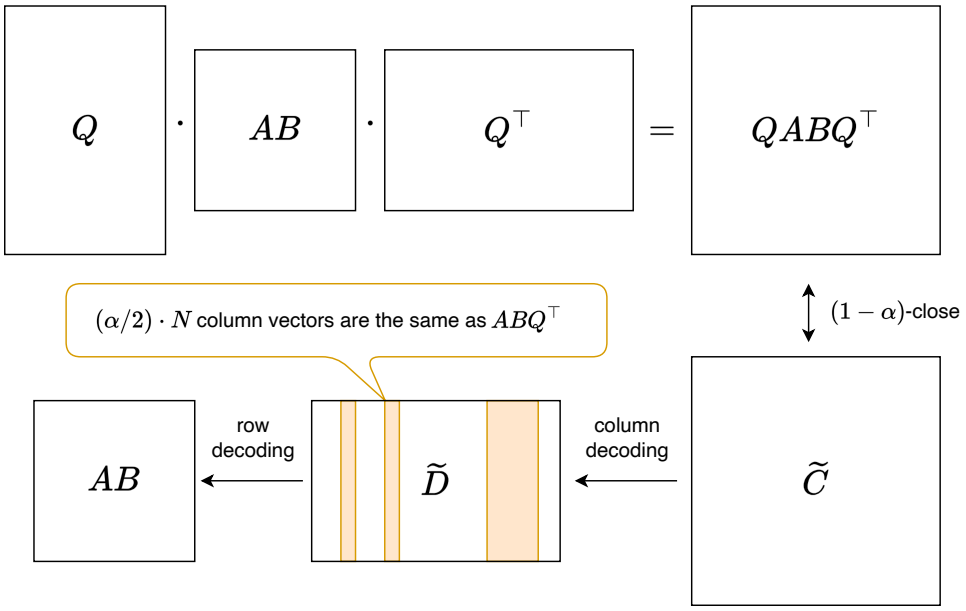

Figure 2: Outline of the reduction of Lemma 4.21. In the first step, we decode each column vector of $\widetilde{C}$. For the resulting matrix $\widetilde{D}$, we decode each of the row vector.

column index $j \in [N]$. Furthermore, this vector d_j can be identified by an efficient randomized verification algorithm (Lemma 4.20). This allows us to compute $(\alpha/2) \cdot N$ column vectors of $ABQ^\top$. Taking the transposition, for all $i \in [n]$, we have a vector $\tilde{f}_i \in \mathbb{F}_p^N$ that is $(1-\alpha/2)$-close to the i-th column vector $f_i \in \mathbb{F}_p^N$ of $Q \cdot (AB)^\top \in \mathbb{F}_p^{N \times n}$. Note that the vector f_i is the encoding of the i-th column vector of $(AB)^\top$. Thus, by using the list-decoding algorithm on input $\tilde{f}_i$ combined with an efficient verification algorithm for all $i \in [n]$, we obtain $(AB)^\top$.

2.3.3 Concrete Codes

In our reduction, we need an encoding that is encodable and list-decodable in nearly linear time over alphabet $\mathbb{F}_p$.

For the case of large field (field of order $\Theta(n)$), we use the Reed–Solomon code, which can be encoded in $\widetilde{O}(n)$ arithmetic operations by using the fast multipoint evaluation [GG13] and is list-decodable within radius $1-\alpha$ if $\alpha > \sqrt{n/N}$ by $\widetilde{O}(N)$ arithmetic operations [Ale05]. Moreover, the list size is at most $O(1/\alpha)$. It is not hard to see that Reed–Solomon codes have a full-rank partition, which can be constructed in linear time. See Section 4.1 for details.

In the case of small field, we use an encoding based on Ta-Shma's construction [Ta-17] (attributed to Rozenman and Wigderson), which is the composition of a derandomized direct sum encoding and a uniquely decodable encoding.

More concretely, we use the encoding $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ that is the composition of two encodings $\mathsf{Enc}_0\colon \mathbb{F}_p^n \to \mathbb{F}_p^{n'}$ and $\mathsf{Enc}_1\colon \mathbb{F}_p^{n'} \to \mathbb{F}_p^N$ defined as follows: Let $n \le n' \le N$ and $\mathsf{Enc}_0\colon \mathbb{F}_p^n \to \mathbb{F}_p^{n'}$ be any encoding that is uniquely decodable within radius ρ_0 for some constant $\rho_0 < 1$. Let $G = ([n'], E)$

be a d-regular expander graph for some $d = O(1)$ and $W \subseteq [n']^k$ be the set of all walks on G of length $k-1$. Set $N = n \cdot d^{k-1}$. We order the set W in the lexicographic order. For $x \in \mathbb{F}_p^n$ and $i \in [N]$, the i-th symbol of the codeword $y = \mathsf{Enc}(x) \in \mathbb{F}_p^N$ is given by

$$y_i = \sum_{j \in [k]} x_{u_j},$$

where $u_1, \dots, u_k \in [n']$ are vertices that the i-th walk in W visits.

Recently, Jeronimo, Srivastava, and Tulsiani [JST21] (for $p = 2$) and Jeronimo [Jer23] presented a nearly linear time decoding algorithm for general class of encodings including the construction above, which deals with radius $1 - 1/p - \varepsilon$ for any constant $\varepsilon > 0$. Thus, the Enc is encodable and list-decodable in nearly linear time within radius $1 - 1/p - \varepsilon$.

Moreover, we prove that the encoding Enc has a full-rank partition if the regular graph G used in the construction has a sufficiently strong expansion property and the girth is more than k. In our application, a random d-regular graph for sufficiently large d suffice. See Section 4.2 for details.

3 Reductions with Preprocessing

The goal of this section is to prove the following theorem.

Theorem 3.1 (the full version of Theorem 1.1)**.** *There exist a randomized polynomial-time algorithm P and a randomized oracle algorithm $M^{\mathcal{O}}$ with the following properties. Let p be a prime number, $\varepsilon, \delta > 0$. For every oracle $\mathcal{O}$ such that*

$$\mathop{\mathbb{E}}_{A,B \sim \mathbb{F}_p^{n \times n}} [\mathrm{dist}(\mathcal{O}(A,B), AB)] \leq 1 - \frac{1}{p} - \varepsilon,$$

the preprocessing algorithm P takes $(n, p, \varepsilon^{-1}, \delta^{-1})$ written in unary as input and, with probability $1-\delta$ over the internal randomness of P, outputs an advice string α such that for every $A, B \in \mathbb{F}_p^{n \times n}$,

$$\Pr_M \big[M^{\mathcal{O}}(A, B; \alpha) = AB \big] \geq 1 - \frac{1}{n},$$

where the probability is over the internal randomness of M. Moreover, $M^{\mathcal{O}}$ runs in time $O(n^2 \log n \cdot \mathsf{poly}(p, 1/\varepsilon, 1/\delta))$ and makes at most $O(\log n) \cdot \widetilde{O}\big(p^2/(\delta^2 \varepsilon^4)\big)$ oracle queries in $\mathbb{F}_p^{n \times n} \times \mathbb{F}_p^{n \times n}$.

We prove this theorem by a sequence of lemmas. Each lemma presents a randomized reduction, which is formalized as a randomized oracle algorithm which is given black-box access to a randomized algorithm. To emphasize that the randomized algorithm is used in a black-box way, we call a randomized algorithm a *randomized oracle*.

Throughout this section, p denotes a prime number and $\mathbb{F}_p$ denotes the field of order p. We say that an algorithm runs in *nearly linear time* if it runs in time $O\big(n^2 \log n \cdot \mathsf{poly}(p, 1/\varepsilon, 1/\delta)\big)$. All the reductions in this section run in nearly linear time.

We may assume without loss of generality that *for every* $(i,j) \in [n]^2$, the (i,j)-th entry of the output of the oracle $\mathcal{O}$ on input (A, B) agrees with $(AB)_{i,j}$ with probability $\frac{1}{p} + \varepsilon$. This can be ensured by using random permutations. A *permutation matrix* Π is a matrix such that there exists a permutation $\pi \colon [n] \to [n]$ such that $\Pi_{i,j} = 1$ if $i = \pi(j)$ and $\Pi_{i,j} = 0$ otherwise.

Lemma 3.2 (implicit in Gola, Shinkar, and Singh [GSS24]). *There exists a randomized linear-time one-query oracle algorithm M such that for every randomized oracle $\mathcal{O}$ such that*

$$q \coloneqq \mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ \mathcal{O}}}[1-\operatorname{dist}(\mathcal{O}(A,B),AB)],$$

it holds that for every $(i,j)\in[n]^2$,

$$q = \Pr_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M}}\left[M^{\mathcal{O}}(A,B)_{i,j} = (AB)_{i,j}\right],$$

where the probability is taken over $A,B\sim\mathbb{F}_p^{n\times n}$ and the internal randomness of M.

Proof. The algorithm $M^{\mathcal{O}}$ operates as follows on input (A,B). It samples uniformly random permutation matrices $\Pi,\Sigma\in\{0,1\}^{n\times n}$, makes a query $(\Pi^{-1}A, B\Sigma^{-1})$ to the oracle $\mathcal{O}$, and outputs $\Pi\cdot\mathcal{O}(\Pi^{-1}A,B\Sigma^{-1})\cdot\Sigma$. This algorithm runs in time $O(n^2)$.

By assumption, we have

$$q = \Pr_{i,j,A,B}[\mathcal{O}(A,B)_{i,j} = (AB)_{i,j}],$$

where the probability is taken over $(i,j)\sim[n]^2$ and $A,B\sim\mathbb{F}_p^{n\times n}$. This is equivalent to saying that for every fixed $(i,j)\in[n]^2$,

$$q = \Pr_{\pi,\sigma,A,B}\left[\mathcal{O}(A,B)_{\pi(i),\sigma(j)} = (AB)_{\pi(i),\sigma(j)}\right],$$

for uniformly random permutations $\pi,\sigma\colon[n]\to[n]$. Using uniformly random permutation matrices Π,Σ, this is also equivalent to

$$q = \Pr_{\Pi,\Sigma,A,B}[(\Pi\cdot\mathcal{O}(A,B)\cdot\Sigma)_{i,j} = (\Pi\cdot AB\cdot\Sigma)_{i,j}].$$

Define $A'\coloneqq\Pi A$ and $B'\coloneqq B\Sigma$. For every fixed Π,Σ, the distribution $(A,B)\sim(\mathbb{F}_p^{n\times n})^2$ induces the same distribution $(A',B')\sim(\mathbb{F}_p^{n\times n})^2$. Thus, we obtain

$$q = \Pr_{\Pi,\Sigma,A',B'}\left[(\Pi\cdot\mathcal{O}(\Pi^{-1}A',B'\Sigma^{-1})\cdot\Sigma)_{i,j} = (A'B')_{i,j}\right],$$

which is equivalent to

$$q = \Pr_{M,A',B'}\left[M^{\mathcal{O}}(A',B')_{i,j} = (A'B')_{i,j}\right].$$

□

By combining Lemma 3.2 with the self-correction algorithm of Blum, Luby, and Rubinfeld [BLR93], if the approximation error of a matrix multiplication algorithm is sufficiently smaller than $\frac{1}{8}$, the error can be corrected. For a technical reason, we state it for rectangular matrix multiplication.

Corollary 3.3 (Gola, Shinkar, and Singh [GSS24]). *There exists a randomized nearly-linear-time $O(\log N)$-query oracle algorithm M such that for every randomized oracle $\mathcal{O}$ such that*

$$\mathop{\mathbb{E}}_{\substack{A\sim\mathbb{F}_p^{N\times n}\\ B\sim\mathbb{F}_p^{n\times N}\\ \mathcal{O}}}[\operatorname{dist}(\mathcal{O}(A,B),AB)]\le\frac{1}{9},$$

it holds that for every $(A, B) \in \mathbb{F}_p^{N\times n} \times \mathbb{F}_p^{n\times N}$,

$$\Pr_{M^{\mathcal{O}}}\left[M^{\mathcal{O}}(A,B) = AB\right] \geq 1 - \frac{1}{N^2},$$

where the probability is taken over the internal randomness of $M^{\mathcal{O}}$.

Proof Sketch. As in the proof of Lemma 3.2, we may assume without loss of generality that an oracle $\mathcal{O}$ satisfies

$$\Pr_{\substack{A\sim\mathbb{F}_p^{N\times n}\\ B\sim\mathbb{F}_p^{n\times N}\\ \mathcal{O}}}[\mathcal{O}(A,B)_{i,j} = (AB)_{i,j}] \geq \frac{8}{9}$$

for every $(i,j) \in [N]^2$. The algorithm $M^{\mathcal{O}}$ repeatedly samples $(R,S) \sim \mathbb{F}_p^{N\times n} \times \mathbb{F}_p^{n\times N}$ and computes $Q := \mathcal{O}(A+R, B+S) - \mathcal{O}(A+R, S) - \mathcal{O}(R, B+S) + \mathcal{O}(R,S)$, and outputs the majority vote of $O(\log N)$ independent trials. □

Corollary 3.3 naturally sets our goal to design an algorithm that approximates matrix multiplication with error at most $\frac{1}{9}$. We construct such an algorithm by using an "embedding reduction," which underlies the proof of Yao's XOR lemma. In order for the embedding reduction to be successful, we need to ensure that for each entry, the most frequent output of an algorithm is the correct value of matrix multiplication. This is achieved by the next lemma.[5]

Lemma 3.4. *There exists a randomized nearly-linear-time one-query oracle algorithm* M *such that for every* $(i,j) \in [n]^2$, *every* $\varepsilon > 0$, *and every randomized oracle* $\mathcal{O}$ *such that*

$$\Pr_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ \mathcal{O}}}[\mathcal{O}(A,B)_{i,j} = (AB)_{i,j}] = \frac{1}{p} + \varepsilon,$$

it holds that for every $d \in \mathbb{F}_p$,

$$\Pr_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M}}\left[M^{\mathcal{O}}(A,B)_{i,j} = (AB)_{i,j} + d\right] = \begin{cases} \frac{1}{p} + \varepsilon & \text{if } d = 0, \\ \frac{1}{p} - \frac{\varepsilon}{p-1} & \text{if } d \neq 0. \end{cases}$$

where the probability is taken over $A, B \sim \mathbb{F}_p^{n\times n}$ *and the internal randomness of* M.

Proof. Let $M^{\mathcal{O}}$ be an oracle algorithm that takes $(A,B) \in (\mathbb{F}_p^{n\times n})^2$ as input, samples $c \sim \mathbb{F}_p \setminus \{0\}$ uniformly and randomly, makes a query $(c\cdot A, B)$ to the oracle $\mathcal{O}$, and outputs $c^{-1}\cdot\mathcal{O}(c\cdot A, B)$.

Fix arbitrary $(i,j) \in [n]^2$ and $d \in \mathbb{F}_p$. Observe that

$$\begin{aligned}
&\Pr_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M}}\left[M^{\mathcal{O}}(A,B)_{i,j} = (AB)_{i,j} + d\right] \\
&= \Pr_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ c\sim\mathbb{F}_p\setminus\{0\},\mathcal{O}}}\left[c^{-1}\cdot\mathcal{O}(cA,B)_{i,j} = (AB)_{i,j} + d\right] \\
&= \Pr_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ c\sim\mathbb{F}_p\setminus\{0\},\mathcal{O}}}[\mathcal{O}(cA,B)_{i,j} = (cAB)_{i,j} + cd] \\
&= \Pr_{\substack{A',B\sim\mathbb{F}_p^{n\times n}\\ c\sim\mathbb{F}_p\setminus\{0\},\mathcal{O}}}\left[\mathcal{O}(A',B)_{i,j} = (A'B)_{i,j} + cd\right]. \qquad (1)
\end{aligned}$$

[5] If $p = 2$, the reduction of Lemma 3.4 is not necessary.

Here, in the last equality, we defined $A' := cA$ and used the fact that for each $c \in \mathbb{F}_p \setminus \{0\}$, the distribution of $A \sim \mathbb{F}_p^{n\times n}$ is identical to that of $cA \sim \mathbb{F}_p^{n\times n}$. If $d = 0$, the probability of (1) is exactly equal to $\frac{1}{p} + \varepsilon$ by assumption. If $d \neq 0$, the probability of (1) is equal to

$$\mathop{\mathbb{E}}_{A',B\sim\mathbb{F}_p^{n\times n},\mathcal{O}}\left[\Pr_{c\sim\mathbb{F}_p\setminus\{0\}}\left[c = d^{-1}\cdot\left(\mathcal{O}(A',B)_{i,j} - (A'B)_{i,j}\right)\right]\right]. \tag{2}$$

We analyze this by considering two cases: Whether the right-hand side $d^{-1}\cdot(\mathcal{O}(A',B)_{i,j} - (A'B)_{i,j})$ is 0 or not. The right-hand side is 0 with probability $\frac{1}{p} + \varepsilon$ and is non-zero with probability $1 - \left(\frac{1}{p} + \varepsilon\right)$ (over the random choice of (A', B) and $\mathcal{O}$). In the former case, the equality does not hold for every $c \in \mathbb{F}_p \setminus \{0\}$. In the latter case, the probability that the equality holds is $\frac{1}{p-1}$ because c is uniformly chosen from $\mathbb{F}_p \setminus \{0\}$. Thus, (2) is equal to

$$\left\{1 - \left(\frac{1}{p} + \varepsilon\right)\right\}\cdot\frac{1}{p-1} = \frac{1}{p} - \frac{\varepsilon}{p-1}.$$

□

Now, we are ready to present what can be proved by the embedding reduction. Assuming that the most frequent output of an oracle $\mathcal{O}$ is correct, we construct a randomized average-case algorithm $M^{\mathcal{O}}$ for matrix multiplication with small error that takes Trevisan–Vadhan advice [TV07], i.e., an advice string that may depend on the internal randomness of M.

Lemma 3.5. *There exists a deterministic nearly-linear-time $O(\log(1/\delta))/\varepsilon^2$-query oracle algorithm M with an advice function α such that for every $n, N \in \mathbb{N}, \delta, \varepsilon > 0$ such that $N/n \geq O(\log(1/\varepsilon))/(\delta\varepsilon)^2$, for every $(i,j) \in [N]^2$ and every randomized oracle $\mathcal{O}$ such that for every $d \in \mathbb{F}_p \setminus \{0\}$,*

$$\Pr_{\substack{\bar{A},\bar{B}\sim\mathbb{F}_p^{N\times N}\\ \mathcal{O}}}\left[\mathcal{O}(\bar{A},\bar{B})_{i,j} = (\bar{A}\bar{B})_{i,j}\right] - \Pr_{\substack{\bar{A},\bar{B}\sim\mathbb{F}_p^{N\times N}\\ \mathcal{O}}}\left[\mathcal{O}(\bar{A},\bar{B})_{i,j} = (\bar{A}\bar{B})_{i,j} + d\right] \geq \varepsilon,$$

it holds that

$$\Pr_{\substack{r\\ A\sim\mathbb{F}_p^{N\times n}\\ B\sim\mathbb{F}_p^{n\times N}}}\left[M^{\mathcal{O}}(A,B;r,\alpha(r))_{i,j} = (AB)_{i,j}\right] \geq 1 - 2p\delta,$$

where the probability is taken over the randomness r of M (a uniformly random string) and $A \sim \mathbb{F}_p^{N\times n}, B \sim \mathbb{F}_p^{n\times N}$. Moreover, the advice function α can be computed in polynomial time.

To analyze the embedding reduction, we use the concentration inequality which underlies direct product theorems.

Lemma 3.6 (Direct product lemma; [IJK09; HS23; HS24])**.** *Let D be a set. For all sufficiently small $\varepsilon > 0$, for every $K \geq O(\log(1/\varepsilon))/(\delta\varepsilon)^2$, for every function $S\colon D^K \to [-1,1]$, it holds that*

$$\Pr_{x\sim D}\left[\left|\mathop{\mathbb{E}}_{y\sim\Gamma(x)}[S(y)] - \mathop{\mathbb{E}}_{y\sim\mathcal{D}^K}[S(y)]\right| \leq \frac{\varepsilon}{2}\right] \geq 1 - \delta.$$

Here, $\Gamma(x)$ is the distribution over $y' \in D^K$ defined by the following sampling procedure: Sample $y \sim D^K$, $k \sim [K]$, replace the k-th element of y with x to obtain y', and output y'.

Proof Sketch. By Hoeffding's inequality, for every $S' \colon D \to [0,1]$, it holds that for every $\alpha \geq 0$,

$$\Pr_{y \sim D^K}\left[\left|\mathop{\mathbb{E}}_{k \sim [K]}\left[S'(y_k)\right] - \mathop{\mathbb{E}}_{x \sim D}\left[S'(x)\right]\right| \leq \alpha\right] \geq 1 - 2\exp(-2\alpha^2 n),$$

where $y = (y_1, \cdots, y_k)$. By the exchange lemma ([HS24, Lemma 3.12]), this concentration inequality implies the concentration inequality of Lemma 3.6, provided that $2\exp(-2\alpha^2 n) \leq \varepsilon/2$ and $\alpha \leq \delta\varepsilon/8$.[6] We choose $\alpha \coloneqq \delta\varepsilon/8$. □

Proof of Lemma 3.5. Let $K \coloneqq N/n$. We may assume without loss of generality that $K \in \mathbb{N}$. Let $D \coloneqq \mathbb{F}_p^{N \times n} \times \mathbb{F}_p^{n \times N}$. We may regard D^K as $\mathbb{F}_p^{N \times N} \times \mathbb{F}_p^{N \times N}$ because $N = Kn$.

The algorithm of $M^{\mathcal{O}}$ is described in Algorithm 1. In one iteration of the algorithm, it samples $(\bar{A}, \bar{B}) \sim \Gamma(A, B)$, takes advice $\alpha = \bar{A}\bar{B} - AB \in \mathbb{F}_p^{N \times N}$ (which can be defined independent of the input (A, B)), queries the oracle $\mathcal{O}$ with $(\bar{A}, \bar{B})$, and computes $\mathcal{O}(\bar{A}, \bar{B}) - \alpha \in \mathbb{F}_p^{N \times N}$. Finally, the algorithm outputs the most frequent element in $\mathbb{F}_p$ for each entry $(i, j) \in [N]^2$. See Algorithm 1 for a more precise description.

Algorithm 1 The description of the embedding reduction $M^{\mathcal{O}}$

Input: $(A, B) \in \mathbb{F}_p^{N \times n} \times \mathbb{F}_p^{n \times N}$, randomness r, advice string $\alpha = (\alpha^{(1)}, \cdots, \alpha^{(T)})$, parameter $T \coloneqq O(\log(1/\delta))/\varepsilon^2$.

1: Initialize a multiset $R_{i,j} \coloneqq \emptyset$ over $\mathbb{F}_p$ for each $(i, j) \in [N]^2$.
2: **for** each $t \in \{1, \cdots, T\}$ **do**
3: Using randomness r, sample $k \sim [K]$ and $(A_\ell, B_\ell) \sim \mathbb{F}_p^{N \times n} \times \mathbb{F}_p^{n \times N}$ for each $\ell \in [K] \setminus \{k\}$.
4: Define $A_k \coloneqq A$ and $B_k \coloneqq B$.
5: Define the matrices:

$$\bar{A} \coloneqq \begin{bmatrix} A_1 & \cdots & A_K \end{bmatrix} \in \left(\mathbb{F}_p^{N \times n}\right)^{1 \times K}, \quad \bar{B} \coloneqq \begin{bmatrix} B_1 \\ \vdots \\ B_K \end{bmatrix} \in \left(\mathbb{F}_p^{n \times N}\right)^{K \times 1}.$$

6: Query the oracle $\mathcal{O}$ with $(\bar{A}, \bar{B})$.
7: **for** each $(i, j) \in [N]^2$ **do**
8: Insert the element $\mathcal{O}(\bar{A}, \bar{B})_{i,j} - \alpha_{i,j}^{(t)} \in \mathbb{F}_p$ to $R_{i,j}$.
▷ We expect $\alpha^{(t)} = \sum_{\ell \in [K] \setminus \{k\}} A_\ell B_\ell \in \mathbb{F}_p^{N \times N}$.
9: **end for**
10: **end for**
11: For each $(i, j) \in [N]^2$, let $Q_{i,j} \in \mathbb{F}_p$ be the element in $\mathbb{F}_p$ that appears most frequently in $R_{i,j}$.

Output: $Q = (Q_{i,j})_{(i,j) \in [N]^2}$.

Fix arbitrary $(i, j) \in [N]^2$ and $d \in \mathbb{F}_p \setminus \{0\}$. Let $S_{i,j}^d \colon D^K \to [-1, 1]$ be the function defined as

$$S_{i,j}^d(\bar{A}, \bar{B}) \coloneqq \Pr_{\mathcal{O}}\left[\mathcal{O}(\bar{A}, \bar{B})_{i,j} = (\bar{A}\bar{B})_{i,j}\right] - \Pr_{\mathcal{O}}\left[\mathcal{O}(\bar{A}, \bar{B})_{i,j} = (\bar{A}\bar{B})_{i,j} + d\right]$$

[6] [HS24, Lemma 3.12] deals with a function whose codomain is $[0, 1]$, whereas we consider a function whose codomain $[-1, 1]$. This discrepancy can be absorbed into constant factors in ε and δ.

for every $(\bar{A}, \bar{B}) \in (\mathbb{F}_p^{N\times n})^{1\times K} \times (\mathbb{F}_p^{n\times N})^{K\times 1}$. By Lemma 3.6, with probability at least $1-\delta$ over $(A,B) \sim \mathbb{F}_p^{N\times n} \times \mathbb{F}_p^{n\times N}$, we obtain

$$\mathop{\mathbb{E}}_{(\bar{A},\bar{B})\sim\Gamma(A,B)}\left[S_{i,j}^d(\bar{A},\bar{B})\right] \geq \mathop{\mathbb{E}}_{\bar{A},\bar{B}}\left[S_{i,j}^d(\bar{A},\bar{B})\right] - \frac{\varepsilon}{2} \geq \frac{\varepsilon}{2},$$

where the last inequality holds by assumption. Define

$$G \coloneqq \left\{(A,B) \in \mathbb{F}_p^{N\times n} \times \mathbb{F}_p^{n\times N} \;\middle|\; \mathop{\mathbb{E}}_{(\bar{A},\bar{B})\sim\Gamma(A,B)}\left[S_{i,j}^d(\bar{A},\bar{B})\right] \geq \frac{\varepsilon}{2} \text{ for every } d \in \mathbb{F}_p \setminus \{0\}\right\}.$$

By taking the union bound over all $d \in \mathbb{F}_p \setminus \{0\}$, we have

$$\Pr_{(A,B)\sim\mathbb{F}_p^{N\times n}\times\mathbb{F}_p^{n\times N}}[(A,B) \in G] \geq 1 - p\delta. \tag{3}$$

We claim that for every $(A,B) \in G$,

$$\Pr_r\left[M^{\mathcal{O}}(A,B;r,\alpha(r))_{i,j} = (AB)_{i,j}\right] \geq 1 - p\delta \tag{4}$$

for some advice function $\alpha(r)$. The randomness $r = (r^{(1)}, \cdots, r^{(T)})$ consists of elements

$$r^{(t)} = \left(k^{(t)}, \left((A_\ell^{(t)}, B_\ell^{(t)}) \;\middle|\; \ell \in [K] \setminus \left\{k^{(t)}\right\}\right)\right),$$

where $r^{(t)}$ denotes the samples drawn in Line 3 of Algorithm 1. We define the advice string $\alpha^{(t)}(r) \coloneqq \sum_{\ell\in[K]\setminus\{k\}} A_\ell^{(t)} B_\ell^{(t)} \in \mathbb{F}_p^{N\times N}$ for every $t \in [T]$. Then, letting $\bar{A}^{(t)}$ and $\bar{B}^{(t)}$ denote the values of $\bar{A}$ and $\bar{B}$, respectively, in the t-th iteration of $M^{\mathcal{O}}$, we have

$$\bar{A}^{(t)}\bar{B}^{(t)} = \sum_{\ell\in[K]} A_\ell^{(t)} B_\ell^{(t)} = AB + \alpha^{(t)}.$$

For every $d \in \mathbb{F}_p$, it follows that $\mathcal{O}(\bar{A}^{(t)}, \bar{B}^{(t)})_{i,j} = (\bar{A}^{(t)}\bar{B}^{(t)})_{i,j} + d$ if and only if $\left(\mathcal{O}(\bar{A}^{(t)}, \bar{B}^{(t)}) - \alpha^{(t)}\right)_{i,j} = (AB)_{i,j} + d$. Thus, for each $d \in \mathbb{F}_p$, the probability that $(AB)_{i,j} + d$ is inserted into $R_{i,j}$ is equal to

$$q_d \coloneqq \Pr_{\substack{(\bar{A},\bar{B})\sim\Gamma(A,B)\\ \mathcal{O}}}\left[\mathcal{O}(\bar{A},\bar{B})_{i,j} = (\bar{A}\bar{B})_{i,j} + d\right].$$

Fix arbitrary $(A,B) \in G$. By the definition of G, for every $d \in \mathbb{F}_p \setminus \{0\}$, we have

$$q_0 - q_d \geq \frac{\varepsilon}{2}. \tag{5}$$

Let $\hat{q}_d \in [0,1]$ be the fraction of $t \in [T]$ such that $(AB)_{i,j} + d$ is inserted into $R_{i,j}$ in the t-th iteration. Since we take $T = O(\log(1/\delta))/\varepsilon^2$ independent samples, Hoeffding's inequality implies that $\hat{q}_d \in [q_d - \frac{\varepsilon}{8}, q_d + \frac{\varepsilon}{8}]$ with probability at least $1-\delta$ over the randomness of $M^{\mathcal{O}}$. By the union bound over all $d \in \mathbb{F}_p$, with probability at least $1 - p\delta$, it holds that $\hat{q}_d \in [q_d - \frac{\varepsilon}{8}, q_d + \frac{\varepsilon}{8}]$ for every $d \in \mathbb{F}_p$, Under this event, we have $Q_{i,j} = (AB)_{i,j}$ because for every $d \in \mathbb{F}_p \setminus \{0\}$,

$$\hat{q}_0 - \hat{q}_d \geq \left(q_0 - \frac{\varepsilon}{8}\right) - \left(q_d + \frac{\varepsilon}{8}\right) \geq \frac{\varepsilon}{2} - \frac{\varepsilon}{4} = \frac{\varepsilon}{4},$$

where the last inequality follows from (5). This completes the proof of (4).

The lemma readily follows from (3) and (4). □

We are now ready to complete the proof of Theorem 3.1.

Proof of Theorem 3.1. Let $\mathcal{O}\colon \mathbb{F}_p^{N\times N} \times \mathbb{F}_p^{N\times N} \to \mathbb{F}_p^{N\times N}$ be an oracle such that

$$\mathop{\mathbb{E}}_{A,B\sim\mathbb{F}_p^{N\times N}}[\mathrm{dist}(\mathcal{O}(A,B),AB)] \le 1-\frac{1}{p}-\varepsilon.$$

Applying Lemma 3.2 to the oracle $\mathcal{O}$, we obtain a randomized algorithm $M_1^{\mathcal{O}}$ such that for every $(i,j)\in[N]^2$,

$$\Pr_{\substack{A,B\sim\mathbb{F}_p^{N\times N}\\ M_1}}\left[M_1^{\mathcal{O}}(A,B)_{i,j} = (AB)_{i,j}\right] \ge \frac{1}{p}+\varepsilon.$$

Applying Lemma 3.4 to the oracle $M_1^{\mathcal{O}}$, we obtain a randomized algorithm $M_2^{\mathcal{O}}$ such that for every $(i,j)\in[N]^2$ and every $d\in\mathbb{F}_p\setminus\{0\}$,

$$\begin{aligned}&\Pr_{\substack{A,B\sim\mathbb{F}_p^{N\times N}\\ M_2^{\mathcal{O}}}}\left[M_2^{\mathcal{O}}(A,B)_{i,j} = (AB)_{i,j}\right] - \Pr_{\substack{A,B\sim\mathbb{F}_p^{N\times N}\\ M_2^{\mathcal{O}}}}\left[M_2^{\mathcal{O}}(A,B)_{i,j} = (AB)_{i,j}+d\right]\\ &\ge \left(\frac{1}{p}+\varepsilon\right) - \left(\frac{1}{p}-\frac{\varepsilon}{p-1}\right) \ge \varepsilon,\end{aligned}$$

We choose $n,K\in\mathbb{N}$ so that $K := N/n = \Theta(\log(1/\varepsilon))/(\delta\varepsilon)^2$. Applying Lemma 3.5 to the oracle $M_2^{\mathcal{O}}$, we obtain a deterministic algorithm $M_3^{\mathcal{O}}$ such that for every $(i,j)\in[N]^2$,

$$\Pr_{\substack{r\\ A\sim\mathbb{F}_p^{N\times n}\\ B\sim\mathbb{F}_p^{n\times N}}}\left[M_3^{\mathcal{O}}(A,B;r,\alpha(r))_{i,j} = (AB)_{i,j}\right] \ge 1-2p\delta.$$

Taking the expectation over all $(i,j)\in[N]^2$, we have

$$\Pr_{\substack{r\\ A\sim\mathbb{F}_p^{N\times n}\\ B\sim\mathbb{F}_p^{n\times N}\\ i,j\sim[N]}}\left[M_3^{\mathcal{O}}(A,B;r,\alpha(r))_{i,j} = (AB)_{i,j}\right] \ge 1-2p\delta.$$

Let $\delta' := 18p\delta$. By Markov's inequality, with probability at least $1-\delta'$ over a random choice of r, it holds that

$$\Pr_{\substack{A\sim\mathbb{F}_p^{N\times n}\\ B\sim\mathbb{F}_p^{n\times N}\\ i,j\sim[N]}}\left[M_3^{\mathcal{O}}(A,B;r,\alpha(r))_{i,j} = (AB)_{i,j}\right] \ge \frac{8}{9}. \tag{6}$$

Define G to be the set of r that satisfies this event; in other words, $r\in G$ if and only if

$$\mathop{\mathbb{E}}_{\substack{A\sim\mathbb{F}_p^{N\times n}\\ B\sim\mathbb{F}_p^{n\times N}}}\left[\mathrm{dist}(M_3^{\mathcal{O}}(A,B;r,\alpha(r)),AB)\right] \le \frac{1}{9}.$$

Fix an arbitrary $r\in G$. Applying Corollary 3.3 to the oracle $M_3^{\mathcal{O}}(\text{-},\text{-};r,\alpha(r))$, we obtain an algorithm $M_4^{\mathcal{O}}$ such that for every $(A,B)\in\mathbb{F}_p^{N\times n}\times\mathbb{F}_p^{n\times N}$,

$$\Pr_{M_4^{\mathcal{O}}}\left[M_4^{\mathcal{O}}(A,B;r,\alpha(r)) = AB\right] \ge 1-\frac{1}{N^2}. \tag{7}$$

We define the final algorithm $M^{\mathcal{O}}$ as follows. It takes $(\bar{A}, \bar{B}) \in \left(\mathbb{F}_p^{N\times N}\right)^2$ as input and an advice string β, partitions A and B so that

$$\bar{A} \coloneqq \begin{bmatrix} A_1 & \cdots & A_K \end{bmatrix} \in \left(\mathbb{F}_p^{N\times n}\right)^{1\times K}, \quad \bar{B} \coloneqq \begin{bmatrix} B_1 \\ \vdots \\ B_K \end{bmatrix} \in \left(\mathbb{F}_p^{n\times N}\right)^{K\times 1},$$

and outputs $\sum_{\ell\in[K]} M_4^{\mathcal{O}}(A_\ell, B_\ell; \beta)$.

The preprocessing algorithm P chooses a random r, computes $\alpha(r)$ and outputs $\beta \coloneqq (r, \alpha(r))$ as an advice string. By (6), with probability at least $1-\delta'$, the algorithm P finds $r \in G$, in which case (7) holds.

We prove the correctness of $M^{\mathcal{O}}(\bar{A}, \bar{B}; \beta)$, where $\beta = (r, \alpha(r))$ for some $r \in G$. By (7) and the union bound over all $\ell \in [K]$, with probability at least $1 - K/N^2 \geq 1 - 1/N$ over the internal randomness of $M_4^{\mathcal{O}}$, it holds that $M_4^{\mathcal{O}}(A_\ell, B_\ell; \beta) = A_\ell B_\ell$ for every $\ell \in [K]$, in which case the output of $M^{\mathcal{O}}$ is equal to $\bar{A}\bar{B} = \sum_{k\in[K]} A_\ell B_\ell$.

The number of the queries of $M^{\mathcal{O}}$ is bounded by

$$K \cdot T \cdot O(\log n) = \frac{\Theta(\log(1/\varepsilon))}{(\delta\varepsilon)^2} \cdot \frac{O(\log(1/\delta))}{\varepsilon^2} \cdot O(\log n) = O\left(\frac{p^2 \log n \log(1/\varepsilon) \log(p/\delta')}{\delta'^2\varepsilon^4}\right),$$

where T is the parameter of Algorithm 1. □

4 Error-Correcting Codes and Full-Rank Partition

Unless otherwise noted, a vector $x \in \mathbb{F}_p^n$ is treated as a column vector. For an index subset $I = \{i_1, \ldots, i_\ell\} \subseteq [n]$ for $i_1 < \cdots < i_\ell$, we denote by $x|_I \in \mathbb{F}_p^{|I|}$ the subvector defined by $x|_I = (x_{i_1}, \ldots, x_{i_\ell})^\top$. For a matrix $A \in \mathbb{F}_p^{m\times n}$ and $I \subseteq [m], J \subseteq [n]$, we denote by $A|_{I,J} \in \mathbb{F}_p^{|I|\times|J|}$ the submatrix of A that consists of $A_{i,j}$ for all $i \in I$ and $j \in J$. We use the shorthand notation $A|_I = A|_{I,[n]}$. We often use the following standard variant of Markov's inequality.

Lemma 4.1. *Let X be a $[0,1]$-valued random variable with mean $\mu > 0$. Then, for any $0 \leq r \leq \mu$,*

$$\Pr[X \geq \mu - r] \geq \frac{r}{1-\mu+r}.$$

Definition 4.2 (codes)**.** *Let $\mathbb{F}_p$ be a finite field. Let $n, N \in \mathbb{N}$ be parameters such that $n \leq N$. An* encoding *is a linear function $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$. The image $\mathsf{Enc}(\mathbb{F}_p^n) \subseteq \mathbb{F}_p^N$ is called a* code*. A* codeword *is an element of a code. By default, we assume that Enc is an injection.*

In particular, we are interested in encodings that are list-decodable in nearly linear time.

Definition 4.3 (list-decodable codes)**.** *An encoding $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ is* ℓ-list-decodable within radius *r if for any $y \in \mathbb{F}_p^N$, the number of vectors $x \in \mathbb{F}_p^n$ such that $\mathrm{dist}(\mathsf{Enc}(x), y) \leq r$ is at most ℓ. A randomized algorithm that outputs all such vectors with probability $2/3$ (over the internal randomness) is called* list-decoding algorithm*. In particular, if $\ell = 1$, then the encoding is said to be* uniquely decodable within radius *r and the algorithm outputting the unique vector is called* unique-decoding algorithm*.*

We introduce the notion of full-rank partition defined as follows.

Definition 4.4 (Full-Rank Partition)**.** *Let* $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ *be an encoding that can be written as* $\mathsf{Enc}\colon x \mapsto Qx$ *for a matrix* $Q \in \mathbb{F}_p^{N\times n}$*. For* $\delta > 0$ *and* $m \in \mathbb{N}$ *such that* $m \le n$*, an* (m,δ)-full-rank partition *is a partition* $\mathcal{P} = P_1 \sqcup \cdots \sqcup P_a \sqcup P'$ *of* $[N]$ *such that*

- $|P_i| = m$ *for every* $i \in [a]$*.*
- $|P'| \le \delta N$*.*
- *For every* $i \in [a]$*, the submatrix* $Q|_{P_i} \in \mathbb{F}_p^{m\times n}$ *is full rank. That is, the row vectors of* Q *in* $P_i \subseteq [N]$ *are linearly independent.*

We are interested in an encoding $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ with $N = O(n)$ that has an (m,δ)-full-rank partition for $m = \Theta(n)$ and small $\delta > 0$. We present useful properties of full-rank partitions.

Lemma 4.5. *Let* $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ *be an encoding that has an* (m,δ)*-full-rank partition* $\mathcal{P} = P_1 \sqcup \cdots \sqcup P_a \sqcup P'$*. Then, for uniformly random* $X \sim \mathbb{F}_p^n$*, the marginal distribution of* $\mathsf{Enc}(X)|_{P_i}$ *is uniform over* $\mathbb{F}_p^m$ *for every* $i \in [a]$*.*

Proof. Write $\mathsf{Enc}\colon x \mapsto Qx$ for a matrix $Q \in \mathbb{F}_p^{N\times n}$. For any $y \in \mathbb{F}_p^m$, the set $\left\{x \in \mathbb{F}_p^n \colon Qx = y\right\} \subseteq \mathbb{F}_p^n$ is an affine subspace of dimension $n-m$, which has cardinality p^{n-m}; thus $\Pr_{X\sim\mathbb{F}_p^n}[\mathsf{Enc}(X) = y] = p^{-m}$. □

Lemma 4.6. *Let* $m \le n \le n' \le N$ *and* $\delta > 0$*. Let* $\mathsf{Enc}_0\colon \mathbb{F}_p^n \to \mathbb{F}_p^{n'}$ *and* $\mathsf{Enc}_1\colon \mathbb{F}_p^{n'} \to \mathbb{F}_p^N$ *be encodings. Suppose* $\mathsf{Enc}_0, \mathsf{Enc}_1$ *can be written as* $\mathsf{Enc}_i(x) = Q_i x$ *for matrices* $Q_0 \in \mathbb{F}_p^{n'\times n}$ *and* $Q_1 \in \mathbb{F}_p^{N\times n'}$*. If* Enc_1 *has an* (m,δ)*-full-rank partition* $\mathcal{P}$*, then* $\mathcal{P}$ *is also an* (m,δ)*-full-rank partition of* $\mathsf{Enc} := \mathsf{Enc}_1 \circ \mathsf{Enc}_0\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$*.*

Proof. Let $Q = Q_1Q_0 \in \mathbb{F}_p^{N\times n}$ Note that $\mathsf{Enc}(x) = \mathsf{Enc}_1(\mathsf{Enc}_0(x)) = Q_1Q_0x = Qx$. Let $\mathcal{P} = P_1 \sqcup \cdots \sqcup P_a \sqcup P'$ be an (m,δ)-full-rank partition of Enc_1. To prove that $\mathcal{P}$ is also an (m,δ)-full-rank partition of Enc, it suffices to prove that each submatrix $Q|_{P_i}$ is full rank. Fix $i \in [a]$ and consider the set of row indexes P_i in the partition. The submatrix $Q|_{P_i} \in \mathbb{F}_p^{m\times n}$ of Q consisting of row vectors in P_i can be written as the product of $Q_1|_{P_i}$ and Q_0. Since $Q_1|_{P_i} \in \mathbb{F}_p^{m\times n'}$ is full rank and Q_0 is column full rank the matrix $Q|_{P_i} \in \mathbb{F}_p^{m\times n}$ is full rank. □

4.1 Example: Reed–Solomon Codes

We show that any Reed–Solomon encoding has a full-rank partition.

Definition 4.7. *Let* $n, N \in \mathbb{N}$ *be parameters and* $\mathbb{F}_p$ *be a finite field of order* $p \ge N$*. Fix a sequence* $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_N)$ *of* N *distinct elements of* $\mathbb{F}_p$*. The* Reed–Solomon encoding $\mathsf{RS}_{p,n,\boldsymbol{\gamma}}\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ *is defined by, for every* $i \in [N]$*,*

$$\mathsf{RS}_{p,n,\boldsymbol{\gamma}}(c_0, \ldots, c_{n-1})_i = \sum_{j=0}^{n-1} c_j \gamma_i^j.$$

Lemma 4.8. *Let* $\mathbb{F}_p$ *be a finite field. Let* $n, N \in \mathbb{N}$ *be such that* n *divides* N*. Then for any sequence* $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_N)$ *of* N *distinct elements of* $\mathbb{F}_p$*, the Reed–Solomon encoding* $\mathsf{RS}_{p,n,\boldsymbol{\gamma}}$ *has an* $(n,0)$*-full-rank partition. Moreover, we can compute an* $(n,0)$*-full-rank partition in* $O(N)$ *time.*

Proof. We claim that any equipartition $\mathcal{P} = P_1 \sqcup \cdots \sqcup P_a$ (i.e., $|P_1| = \cdots = |P_a| = n = N/a$) of $[N]$ is indeed a $(n, 0)$-full-rank partition (here, we set $P' = \emptyset$). Note that any such partition can be computed in time $O(N)$. By definition, the encoding $\mathsf{RS}_{p,n,\boldsymbol{\gamma}}$ can be written as $\mathsf{RS}_{p,n,\boldsymbol{\gamma}} \colon x \mapsto Qx$ for the Vandermonde matrix

$$Q = \begin{bmatrix} 1 & \gamma_1 & \gamma_1^2 & \dots & \gamma_1^{n-1} \\ 1 & \gamma_2 & \gamma_2^2 & \dots & \gamma_2^{n-1} \\ & & \vdots & & \\ 1 & \gamma_N & \gamma_N^2 & \dots & \gamma_N^{n-1} \end{bmatrix}.$$

For any size-n subset $P \subseteq [N]$, the submatrix $Q|_P$ obtained by concatenating the q-th row vector of Q for all $q \in P$ is again a Vandermonde matrix, and is nonsingular since $(\gamma_q)_{q \in P}$ are distinct. Therefore, these row vectors are linearly independent. □

4.2 Example: Walk-Amplified Codes

In Reed–Solomon codes, the field size p must be $p \geq N$. To deal a finite field $\mathbb{F}_p$ with small p, we invoke the framework of *derandomized direct sum encoding*.

Definition 4.9 (direct sum encoding)**.** *Let $\mathbb{F}_p$ be a finite field, and $k \in \mathbb{N}$. Let $W \subseteq [n]^k$ be a collection of k-tuples and order the elements of W as $W = \{w_1, \dots, w_N\}$. The* direct sum encoding with respect to W *is the encoding $\mathsf{Enc} \colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ defined by, for each $i \in [N]$ and $w_i = (v_1, \dots, v_k) \in W$,*

$$\mathsf{Enc}(x)_i = x_{v_1} + \cdots + x_{v_k}.$$

In this paper, we focus on a direct sum encoding with respect to a collection of $(k-1)$-step random walks on an expander graph.

Definition 4.10 (k-walk collection)**.** *Let $G = ([n], E)$ be an undirected regular graph. The* k-walk collection on G *is the collection of all walks of length $k-1$ on G. That is,*

$$W = \Big\{ (w_1, \dots, w_k) \in [n]^k \colon \{w_i, w_{i+1}\} \in E \text{ for all } i \in [k-1] \Big\}.$$

By default, we order the elements of W in the lexicographic order.

We consider a k-walk collection on a spectral expander graph with high girth.

Definition 4.11 (Expander Graph and Girth)**.** *For a d-regular (multi)graph $G = (V, E)$, let $P \in [0,1]^{V \times V}$ be the normalized adjacency matrix defined by $P_{u,v} = m_{u,v}/d$, where $m_{u,v}$ is the number of edges between u and v. For $\lambda \in [0,1]$, we say that G is* λ-expander *if the eigenvalues $1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{|V|} \geq -1$ of P satisfies $\max\{|\lambda_2|, |\lambda_{|V|}|\} \leq \lambda$. The* girth *of a simple graph G is the minimum length of cycles of G.*

A canonical example of expander graph is a random regular graph. We use the following useful properties of random regular graphs.

Lemma 4.12. *For $n, d \in \mathbb{N}$ such that nd is even, let $G_{n,d}$ be a random d-regular graph, a graph chosen uniformly at random from the set of all vertex-labeled simple d-regular n-vertex graphs. Then, for every fixed $d \geq 3$, the following holds:*

- *If d is even, then there exists an $O(n)$-time randomized algorithm that, on input 1^n, samples $G_{n,d}$ with probability $2/3$ for all large n [Wor99].* [7]
- *There exists a constant $c > 0$ that depends only on d such that $G_{n,d}$ is $\left(\frac{2\sqrt{d-1}}{d} + c\left(\frac{\log\log n}{\log n}\right)^2\right)$-expander with probability $1 - n^{-10}$ [Fri03; Bor15].*
- *For any constant $g \in \mathbb{N}$, with probability $\exp\left(-\sum_{r=3}^{g} \frac{(d-1)^r}{2r} + o(1)\right)$ as $n \to \infty$, the random graph $G_{n,d}$ has girth greater than g [MWW04].*

Jeronimo [Jer23] and Jeronimo, Srivastava, and Tulsiani [JST21] considered a direct sum encoding with respect to W satisfying a certain expansion property which they called *splittability*. They proved that the encoding obtained by composing such direct sum encoding Enc with a uniquely decodable code with constant distance admits a near-linear-time list-decoding algorithm. In this paper, we consider the following encoding.

Definition 4.13 (Walk-Amplified Encoding)**.** *Let $\mathbb{F}_p$ be a finite field. Let $k, d, g \in \mathbb{N}$ and $\lambda > 0$ be parameters. Let $n \leq n' \leq N$ and $W \subseteq [n']^k$ be the k-walk collection on an n'-vertex d-regular λ-expander graph with girth at least g. Consider the direct sum encoding $\mathsf{Enc}_1 \colon \mathbb{F}_p^{n'} \to \mathbb{F}_p^N$ with respect to W. For an encoding $\mathsf{Enc}_0 \colon \mathbb{F}_p^n \to \mathbb{F}_p^{n'}$, a (k, d, λ, g)-*walk-amplified encoding with base encoding Enc_0 *is the encoding $\mathsf{Enc} \colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ given by $\mathsf{Enc} = \mathsf{Enc}_1 \circ \mathsf{Enc}_0 \colon \mathbb{F}_p^n \to \mathbb{F}_p^N$.*

We prove that the walk-amplified encoding has a full-rank partition for a suitable choice of parameters.

Lemma 4.14. *For any given $k \geq 2, \lambda > 0, \delta > 0$ such that $(2\lambda)^k \leq \frac{\delta^2}{288}$, we can choose parameters $m = \Theta(n)$ and $d = O(1)$ such that the following holds: Let $\mathsf{Enc} \colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ be a (k, d, λ, k)-walk-amplified encoding with base encoding $\mathsf{Enc}_0 \colon \mathbb{F}_p^n \to \mathbb{F}_p^{n'}$. Then Enc has an (m, δ)-full-rank partition. Moreover, there exists an $O(N)$-time randomized algorithm that on input the graph used for the encoding outputs an (m, δ)-full-rank partition with probability $2/3$.*

Remark 4.15. *We refer to Eq. (9) for a concrete parameter setting.*

To prove Lemma 4.14, we begin with showing that the direct sum encoding has a full-rank partition.

Lemma 4.16. *For any given $k \geq 2, \lambda > 0, \delta > 0$ such that $(2\lambda)^k \leq \frac{\delta^2}{288}$, we can choose parameters $m = \Theta(n)$ and $d = O(1)$ such that the following holds: Let $\mathsf{Enc} \colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ be a direct sum encoding with respect to a k-walk collection on a d-regular λ-expander graph $G = (V, E)$ with girth at least k, where $N = d^{k-1} \cdot n$. Then Enc has an (m, δ)-full-rank partition. Moreover, there exists an $O(N)$-time randomized algorithm that on input G, m, k, δ, outputs an (m, δ)-full-rank partition with probability $2/3$.*

[7] If the degree d is odd, then we can sample $G_{n,d}$ for every large even n.

To prove Lemma 4.16, we invoke a well-known result that states that the probability that a $(k-1)$-step random walk keeps staying in a fixed subset decays exponentially in k.

Lemma 4.17 ([HLW06, Theorem 3.6])**.** *Let $G = (V, E)$ be an n-vertex d-regular λ-expander graph and $B \subseteq V$ be a vertex subset of density $\beta = |B|/n$. Let $(X_1, \dots, X_k) \in V^k$ be the random walk on G of length $k-1$, starting from a uniformly random vertex $X_1 \sim V$. Then, we have*

$$\Pr[\{X_1, \dots, X_k\} \subseteq B] \le (\lambda + \beta)^k.$$

Proof of Lemma 4.16. Write $\mathsf{Enc}\colon x \mapsto Qx$ for a matrix $Q \in \mathbb{F}_p^{N \times n}$. Let $W \subseteq [n]^k$ be the k-walk collection on the d-regular λ-expander graph G, where $|W| = N = d^{k-1} \cdot n$. Note that the row vector of Q indexed by the walk $w \in W$ is the vector $x \in \mathbb{F}_p^n$ whose v-th entry is the number of times the walk w visits v. We identify $[N]$ as the k-walk collection W. Thus, $P \subseteq [N]$ is a set of k-walks.

For a walk $w = (v_1, \dots, v_k) \in W$, let $\mathsf{visit}(w) = \{v_1, \dots, v_k\} \subseteq V$ denote the set of vertices visited by w. For a subset of walks $P' \subseteq W$, let $\mathsf{visit}(P') = \bigcup_{w \in P'} \mathsf{visit}(w)$. We say that a walk $w \in P'$ *explores* P' if there exists a vertex $v \in \mathsf{visit}(w) \setminus \mathsf{visit}(P' \setminus \{w\})$ such that w visits v *exactly once*.

Given k, δ, λ, we first choose an auxiliary parameter $m' = \Theta(n)$ that will be specified later. In the following, we prove that if $P' \sim \binom{W}{m'}$ is a random set of m' distinct walks, then most walks in P' explore P' with probability $1 - o(1)$.

Claim 4.18. *For any $m' < |W|$, let $P' \sim \binom{W}{m'}$ be a uniformly random size-m' subset of W. Let $X(P')$ be the number of walks in P' that explore P'. Then, for any $\gamma > 0$, we have*

$$\Pr_{P'}\left[X(P') \ge \left\lceil \left(1 - \left(\lambda + \frac{m'k}{n}\right)^k - \frac{k}{d} - \gamma\right) m' \right\rceil\right] \ge 1 - \frac{(\lambda + m'k/n)^k + k/d}{\gamma}.$$

Proof. A walk $w = (v_1, \dots, v_k)$ is said to be *non-backtracking* if for every $i = 2, \dots, k-1$, we have $v_{i-1} \neq v_{i+1}$. Since the girth of the underlying graph G is at least k, if $w = (v_1, \dots, v_k) \in W$ is non-backtracking, then $v_1, \dots, v_k$ are distinct. By the union bound, the random walk $w \sim W$ is non-backtracking with probability at least $1 - k/d$.

Now, consider the expectation

$$\mathop{\mathbb{E}}_{P'}[X(P')] = m' \cdot \Pr_{\substack{P' \sim \binom{W}{m'} \\ w \in P'}}\left[w \text{ explores } P'\right]. \tag{8}$$

Since G is λ-expander and has girth at least k, this quantity satisfies

$$\begin{aligned}
\text{RHS of (8)} &= \mathop{\mathbb{E}}_{P'' \sim \binom{W}{m-1}} \left[\Pr_{w \sim W \setminus P''} \left[w \text{ explores } P'' \cup \{w\} \right] \right] \\
&= \mathop{\mathbb{E}}_{P'' \sim \binom{W}{m-1}} \left[\Pr_{w \sim W} \left[w \text{ explores } P' \mid w \notin P'' \right] \right] \\
&\geq \mathop{\mathbb{E}}_{P'' \sim \binom{W}{m-1}} \left[\Pr_{w \sim W} \left[w \text{ explores } P'' \right] \right] \\
&\geq \mathop{\mathbb{E}}_{P'' \sim \binom{W}{m-1}} \left[\Pr_{w \sim W} \left[\mathsf{visit}(w) \not\subseteq \mathsf{visit}(P'') \text{ and } w \text{ is non-backtracking} \right] \right] \\
&\geq 1 - \left(\lambda + \frac{m'k}{n} \right)^k - \frac{k}{d}. && \because \text{Lemma 4.17}
\end{aligned}$$

Therefore, $\mathbb{E}[X(P')] \geq \left(1 - \left(\lambda + \frac{m'k}{n}\right)^k - \frac{k}{d}\right) \cdot m'$. By Markov's inequality (Lemma 4.1) for $\frac{X(P')}{m'}$, it holds for any $\gamma > 0$ that

$$\Pr\left[\frac{X(P')}{m'} \geq 1 - \left(\lambda + \frac{m'k}{n} \right)^k - \frac{k}{d} - \gamma \right] \geq \frac{\gamma}{(\lambda + m'k/n)^k + k/d + \gamma} \geq 1 - \frac{(\lambda + m'k/n)^k + k/d}{\gamma}.$$

Since $X(P')$ is a nonnegative integer, we obtain the claim. □

We continue the proof of Lemma 4.16. For given k, λ, δ such that $(2\lambda)^k \leq \frac{\delta^2}{288}$, set

$$\begin{aligned}
m' &= \left\lfloor \frac{\lambda n}{k} \right\rfloor, \quad & m &= \left\lceil \left(1 - \frac{\delta}{6}\right) m' \right\rceil, \\
a' &= \left\lfloor \frac{N}{m'} \right\rfloor, \quad & a &= \left\lceil \left(1 - \frac{\delta}{3}\right) a' \right\rceil, \\
d &= \left\lceil \frac{288k}{\delta^2} \right\rceil.
\end{aligned} \tag{9}$$

Let $\mathcal{P}' = P'_1 \sqcup \cdots \sqcup P'_{a'} \sqcup P''$ be a partition of $[N]$ that is chosen uniformly at random from those that satisfy $|P'_i| = m'$ for every $i \in [a']$ and $|P''| = N \bmod m'$. Specifically, $\mathcal{P}' = P'_1 \sqcup \cdots \sqcup P'_{a'} \sqcup P''$ can be obtained by randomly shuffling $[N]$ and then taking the first m' elements as P'_1, the second m' elements as P'_2, and so on and letting P'' be the rest. Note that the marginal distribution of each P'_i is uniform over $\binom{W}{m'}$. Let Y be the number of $i \in [a']$ such that $X(P'_i) \geq m$.

By our parameter choice Eq. (9), we have

$$\begin{aligned}
\left(\lambda + \frac{m'k}{n} \right)^k + \frac{k}{d} &\leq (2\lambda)^k + \frac{\delta^2}{288} && \because m' \leq \frac{\lambda n}{k} \text{ and } d \geq \frac{288k}{\delta^2} \\
&\leq \frac{\delta^2}{144} && \because (2\lambda)^k \leq \frac{\delta^2}{288}
\end{aligned}$$

and thus, for $\gamma = \frac{\delta}{12}$,

$$\left\lceil \left(1 - \left(\lambda - \frac{m'k}{n}\right)^k + \frac{k}{d} - \gamma\right) m' \right\rceil \geq \left\lceil \left(1 - \frac{\delta^2}{144} - \frac{\delta}{12}\right) m' \right\rceil \geq \left\lceil \left(1 - \frac{\delta}{6}\right) m' \right\rceil = m.$$

From Claim 4.18 for $\gamma = \frac{\delta}{12}$, we have

$$\begin{aligned}\Pr_{\mathcal{P}'}\left[X(P_i') \geq m\right] &\geq \Pr_{\mathcal{P}'}\left[X(P_i') \geq \left\lceil \left(1 - \left(\lambda + \frac{m'k}{n}\right)^k + \frac{k}{d} - \gamma\right)m' \right\rceil\right] \\ &\geq 1 - \frac{\left(\lambda + \frac{m'k}{n}\right)^k + \frac{k}{d}}{\gamma} \\ &\geq 1 - \frac{\delta}{12}.\end{aligned}$$

In particular, we have $\mathbb{E}_{\mathcal{P}'}[Y] \geq \left(1 - \frac{\delta}{12}\right)a'$. By Markov's inequality (Lemma 4.1) for $\frac{Y}{a'}$, for $\gamma' = \frac{\delta}{4}$, we have

$$\begin{aligned}\Pr_{\mathcal{P}'}[Y \geq a] &\geq \Pr_{\mathcal{P}'}\left[Y \geq \left(1 - \frac{\delta}{12} - \gamma'\right)a'\right] \\ &\geq 1 - \frac{\delta/12}{\gamma'} \\ &= \frac{2}{3}.\end{aligned}$$

We present a construction of an (m, δ)-full-rank partition $\mathcal{P}$. Suppose $Y \geq a$, which occurs with probability $2/3$ over the choice of $\mathcal{P}'$. Then, there are at least a sets P_i' such that $X(P_i') \geq m$. For simplicity, suppose $P_1', \ldots, P_a'$ be such sets. For every $i \in [a]$, we can take m walks that explore P_i'. Let $P_i \subseteq P_i'$ be the set of such m walks. Let $P' = [N] \setminus \bigcup_{i\in[a]} P_i$. Define a partition $\mathcal{P} = P_1 \sqcup \cdots \sqcup P_a \sqcup P'$.

We consider the construction time of $\mathcal{P}$.

- The random choice of $\mathcal{P}'$ can be done in time $O(N)$.
- For each $P_i' \in \mathcal{P}'$ we can enumerate every walk $w \in P_i'$ that explores $\mathcal{P}'$ in time $O(|P_i'|)$ by counting the total number of times all walks visit v for every $v \in \mathsf{visit}(P_i')$.

Therefore, we can construct $\mathcal{P}$ in time $O(N)$ with probability $2/3$.

We prove the correctness (conditioned on $Y \geq a$). Note that each P_i has cardinality m and

$$\begin{aligned}\left|P'\right| &= N - am \\ &\leq N - \left(1 - \frac{\delta}{6}\right)\left(1 - \frac{\delta}{3}\right)a'm' \\ &\leq N - \left(1 - \frac{\delta}{2}\right)(N - m') && \because a' = \lfloor N/m' \rfloor \\ &\leq \frac{\delta N}{2} + m' \\ &\leq \frac{\delta N}{2} + \frac{\lambda n}{k} \\ &\leq \left(\frac{\delta}{2} + \frac{1}{d}\right)N && \because n = d^{-k+1}N \text{ and } k \geq 2 \\ &\leq \delta N. && \because d \geq 2/\delta\end{aligned}$$

We prove that for every P_i, the row vectors of Q in the row index set P_i are linearly independent. Every walk $w \in P_i$ satisfies $\mathsf{visit}(w) \setminus \mathsf{visit}(P_i \setminus \{w\}) \supseteq \mathsf{visit}(w) \setminus \mathsf{visit}(P_i' \setminus \{w\}) \neq \emptyset$; thus w explores P_i. Consider the row vector $x \in \mathbb{F}_p^n$ of Q that corresponds to the walk $w \in P_i$. Since w explores P_i, the vector x satisfies $x_v = 1$ for some $v \in \mathsf{visit}(w) \setminus \mathsf{visit}(P_i \setminus \{w\})$ (since w visits some v exactly once). Moreover, for all $w' \in P_i \setminus \{w\}$, the corresponding row vector x' satisfies $x'_v = 0$ since w' does not visit v. Thus, x is linearly independent of any other row vectors corresponding walks in $P_i \setminus \{w\}$. Therefore, the partition $\mathcal{P} = P_1 \sqcup \cdots \sqcup P_a \sqcup P'$ is an (m, δ)-full-rank partition conditioned on $Y \geq a$. □

Proof of Lemma 4.14. Let $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ be a (k, d, λ, k)-walk-amplified encoding with base encoding Enc_0. Write $\mathsf{Enc} = \mathsf{Enc}_1 \circ \mathsf{Enc}_0$ for the direct sum encoding Enc_1. From Lemma 4.16, we can compute an (m, δ)-full-rank partition of Enc_1 by an $O(N)$-time randomized algorithm. Moreover, from Lemma 4.6, this full-rank partition is also a full-rank partition of Enc. □

sectionUniform Reductions for Matrix Multiplication We present a worst-case-to-average-case and exact-to-approximation reduction for matrix multiplication. Specifically, we prove Theorems 1.3 and 1.4. Throughout this section, the computational time of an algorithm amounts to the number of arithmetic operations over $\mathbb{F}_p$, which incurs the $\mathsf{polylog}(p)$ overhead in the running time of the reductions of Theorems 1.3 and 1.4 if $p = p(n) = \omega(1)$.

4.3 Left-Right Encoding

A key component of our proof is the *left-right encoding*.

Definition 4.19. *Let* $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ *be an encoding that can be written as* $\mathsf{Enc}\colon x \mapsto Qx$ *for a matrix* $Q \in \mathbb{F}_p^{N \times n}$. *A* left-right encoding with respect to Enc *is an encoding* $\mathsf{Enc}^\star\colon \mathbb{F}_p^{n \times n} \to \mathbb{F}_p^{N \times N}$ *defined by*

$$\mathsf{Enc}^\star(A) = QAQ^\top.$$

In the following, we show how to reconstruct AB given $A, B \in \mathbb{F}_p^{n \times n}$ and $C \in \mathbb{F}_p^{N \times N}$ that is $(1 - \alpha)$-close to $\mathsf{Enc}^\star(AB)$.

First, we present an efficient data structure algorithm that verifies matrix-vector product. The proof follows the idea of [HS23, Lemma 6.2 of the full version].

Lemma 4.20. *There exists a pair of algorithms* $(M_{\mathrm{pre}}, M_{\mathrm{query}})$ *that satisfies the following:*

- *In preprocess phase, on input* $K \in \mathbb{F}_p^{N \times n}$ *and* $L \in \mathbb{F}_p^{n \times N}$, *the randomized algorithm* $M_{\mathrm{pre}}(K, L)$ *runs in time* $O(Nn \log N)$ *and output a string* π.
- *In query phase, given input* $x, y \in \mathbb{F}_p^N$ *and query access to* $\pi = M_{\mathrm{pre}}(K, L)$, *the deterministic oracle algorithm* $M_{\mathrm{query}}^{\pi}(x, y)$ *decides whether* $x = KLy$ *or not correctly with probability* $1 - 1/N^3$ *(over the internal randomness of* M_{pre}*) in time* $O(N \log N)$.

Proof. In the preprocess phase, M_{pre} runs on input $K \in \mathbb{F}_p^{N \times n}$ and $L \in \mathbb{F}_p^{n \times N}$ as follows:

1. Choose random vectors $r_1, \ldots, r_m \sim \{0, 1\}^N$ for some $m = O(\log N)$.
2. For all $j \in [m]$, compute $w_j := r_j^\top KL$.

3. Output the vectors $(w_j)_{j\in[m]}$ and $(r_j)_{j\in[m]}$ as π.

In the query phase, $M^{\pi}_{\mathrm{query}}(x,y)$ for $\pi = M_{\mathrm{pre}}(K,L)$ outputs Yes if and only if $r_j^\top x = w_j y$ for all $j \in [m]$.

Note that the running times of M_{pre} and M_{query} are $O(Nn\log N)$ and $O(N\log N)$, respectively.

We prove the correctness of the query phase. Note that the criterion $r_j^\top x = w_j y$ is equivalent to $r_j^\top x = r_j^\top KLy$. If the input (x,y) satisfies $x = KLy$, then the query algorithm M_{query} outputs Yes with probability 1. Suppose $x \neq KLy$. Then, for such (x,y), we have

$$\Pr_{r_1,\dots,r_m}\left[{}^{\forall} j \in [m],\, r_j^\top x = r_j^\top KLy\right] = \Pr_{r_1,\dots,r_m}\left[{}^{\forall} j \in [m],\, r_j^\top (x - KLy) = 0\right] \le \frac{1}{2^m}.$$

Therefore, M_{query} outputs No with probability at least $1-1/2^m \ge 1-1/N^2$ for some $m = O(\log N)$. □

In the following, we show how to reconstruct $AB \in \mathbb{F}_p^{n\times n}$ given as input $A, B \in \mathbb{F}_p^{n\times n}$ and $\widetilde{C} \in \mathbb{F}_p^{N\times N}$ such that $\mathrm{dist}(\widetilde{C}, \mathsf{Enc}^\star(AB)) \le 1-\alpha$, provided that the encoding Enc used in $\mathsf{Enc}^\star$ is list-decodable within radius $1-\alpha/2$ in nearly linear time.

Lemma 4.21. *Let $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ be an encoding that can be computed in time $\widetilde{O}(n)$ and is ℓ-list decodable within radius $1-\alpha/2$ for $\ell = O_\alpha(1)$ associated with a list decoding algorithm running in time $\ell\cdot\widetilde{O}(N)$.*

Then, there exists a randomized algorithm M that, given $A, B \in \mathbb{F}_p^{n\times n}$ and $\widetilde{C} \in \mathbb{F}_p^{N\times N}$ as input, runs in time $\ell\cdot\widetilde{O}(N^2)$ and outputs AB with probability $1-o(1)$ (over the internal randomness of M) provided that $\mathrm{dist}(\widetilde{C}, \mathsf{Enc}^\star(AB)) \le 1-\alpha$, where $\mathsf{Enc}^\star\colon \mathbb{F}_p^{n\times n} \to \mathbb{F}_p^{N\times N}$ is the left-right encoding with respect to Enc.

Proof. Write $\mathsf{Enc}\colon x \mapsto Qx$ for a matrix $Q \in \mathbb{F}_p^{N\times n}$ and let $\tilde{c}_1, \dots, \tilde{c}_N$ be the column vectors of $\widetilde{C}$. Let $(M_{\mathrm{pre}}, M_{\mathrm{query}})$ be the verification algorithm of Lemma 4.20. The algorithm M consists of two phases: column-decoding and row-decoding phases (Fig. 2).

Column-decoding phase. The column-decoding phase runs as follows: On input $A, B, \widetilde{C}$,

1. Let $\widetilde{D} \in \mathbb{F}_p^{n\times N}$ be a matrix initialized as the all-zero matrix.
2. Compute $BQ^\top$ using the encoding algorithm and then run the preprocess M_{pre} on input (K,L) for $K = A, L = BQ^\top$.
3. For each $j \in [N]$, do the following:
   (a) For each column vector $\tilde{c}_j \in \mathbb{F}_p^N$ of $\widetilde{C}$, run the list decoding algorithm on input $\tilde{c}_j$.
   (b) If the list decoding algorithm[8] outputs a set $Z \subseteq \mathbb{F}_p^n$ of size at most ℓ, for every $z \in Z$, run $M_{\mathrm{query}}(\mathsf{Enc}(z), e_j)$, where $e_j \in \{0,1\}^n$ denotes the indicator vector of index j. If the output is Yes, then replace the j-th column vector of $\widetilde{D}$ by z.
4. Output $\widetilde{D}$.

[8]Here, we repeat the list decoding algorithm for $O(\log N)$ times and output the majority vote to ensure that it succeeds with probability $1-1/N^3$. We do the same in the row-decoding phase.

We claim that the matrix $\widetilde{D}$ agrees with $ABQ^\top$ on at least $(\alpha/2)\cdot N$ columns with high probability.

Claim 4.22. *Let $\tilde{d}_1,\dots,\tilde{d}_N$ be column vectors of $\widetilde{D}$ and $d_1,\dots,d_N$ be column vectors of $D := ABQ^\top$. Then, with probability at least $1-\ell/N^2$ (over the internal randomness of the column-decoding phase), there are at least $(\alpha/2)\cdot N$ indices $j\in[N]$ such that $\tilde{d}_j = d_j$.*

Proof of Claim 4.22. Let $C = \mathsf{Enc}^\star(AB) = QABQ^\top$ and $c_1,\dots,c_N \in \mathbb{F}_p^N$ be its column vectors. Note that $c_j = Qd_j = \mathsf{Enc}(d_j)$. We say that an index $j\in[N]$ is *good* if $\mathrm{dist}(c_j,\tilde{c}_j)\le 1-\alpha/2$. Since $\mathrm{dist}(C,\widetilde{C})\le 1-\alpha$, by Markov's inequality (Lemma 4.1), at least $(\alpha/2)\cdot N$ column indices j are good.

Suppose j is good in the iteration of Step 3. Since $\mathrm{dist}(c_j,\tilde{c}_j)\le 1-\alpha/2$, the output Z of the list decoding algorithm contains d_j. The vector d_j in the list can be identified in Step 3(b) with probability $1-\ell/N^3$ since M_{query} checks if $d_j = KLe_j = De_j$ and by the union bound over $z\in Z$. Finally, by the union bound over good $j\in[N]$, with probability $1-\ell/N^2$, the output $\widetilde{D}$ contains d_j as the j-th column vector for all good $j\in[N]$. □

We continue the proof of Lemma 4.21. Consider the running time of column-decoding phase. Let $T_{\mathrm{enc}}(n) = \widetilde{O}(n)$ be the time needed for computing Enc and $T_{\mathrm{dec}}(N) = \ell\cdot\widetilde{O}(N)$ be the running time of the list-decoding algorithm.

- From Lemma 4.20, Step 2 can be done in time $O(nT_{\mathrm{enc}}(n) + Nn\log N)$.
- Each of the iteration of Step 3 can be done in time $O(T_{\mathrm{dec}}(N)\log N + \ell N\log N)$ (recall that we repeat the list decoding algorithm for $O(\log N)$ times).

Therefore, the running time of the column-decoding phase is at most $\ell\cdot\widetilde{O}(N^2)$.

Row-decoding phase. For the matrix $\widetilde{D}$ obtained by the column-decoding phase, let $\widetilde{F} := \left(\widetilde{D}\right)^\top \in \mathbb{F}_p^{N\times n}$ and $\tilde{f}_1,\dots,\tilde{f}_n\in\mathbb{F}_p^N$ be the column vectors of $\widetilde{F}$. Let $F = D^\top = QAB$ and $f_1,\dots,f_n\in\mathbb{F}_p^N$ be the column vectors of F. Since F is obtained by encoding AB, running the list decoding algorithm on $\tilde{f}_j$ outputs a small list of vectors containing the j-th column vector of AB for every $j\in[n]$. This vector can be identified by Lemma 4.20.

To state it more formally, consider the pair $(M_{\mathrm{pre}}, M_{\mathrm{query}})$ of Lemma 4.20. The row-decoding phase runs on input A, B and $\widetilde{F}$ as follows:

1. Let $\widetilde{G}\in\mathbb{F}_p^{n\times n}$ be a matrix initialized as the all-zero matrix.
2. Run the preprocess M_{pre} on input (A,B).
3. For each $i\in[n]$, do the following:
   (a) For the i-th column vector $\tilde{f}_i\in\mathbb{F}_p^N$ of $\widetilde{F}$, run the list decoding algorithm on input $\tilde{f}_i$, obtaining a set of vectors $Z\subseteq\mathbb{F}_p^n$ of size at most ℓ.
   (b) For each $z\in Z$, run $M_{\mathrm{query}}(z,e_i)$ (recall that e_j is the indicator vector). If the output is Yes, then replace the i-th column vector of $\widetilde{G}$ by z.
4. Output $\widetilde{G}$.

We claim that $G = AB$ with probability at least $1 - \ell/n^2$ conditioned on that $\widetilde{F}$ agrees with $D^\top = QAB$ on at least $(\alpha/2)N$ rows.

From Claim 4.22, for each $i \in [n]$, we have $\mathrm{dist}(f_i, \tilde{f}_i) \le 1 - \alpha/2$. Therefore, in Step 3(a), the list decoding algorithm outputs a set $Z \subseteq \mathbb{F}_p^n$ of size at most ℓ that contains the i-th column vector of AB. Such column vector can be identified at Step 3(b) since the verification query algorithm $M_{\mathrm{query}}(z, e_i)$ checks if $z = ABe_i$. During the repetition of Step 3, we run M_{query} for ℓn times in total. By the union bound of them, we have that M_{query} outputs the correct answer at every call, which yields that $G = AB$ with probability at least $1 - \ell/n^2$.

Finally, consider the running time. Let $T_{\mathrm{enc}}(n) = \widetilde{O}(n)$ be the time needed for computing Enc and $T_{\mathrm{dec}}(N) = \ell \cdot \widetilde{O}(N)$ be the running time of the list-decoding algorithm.

- From Lemma 4.20 for $N = n$, Step 2 can be done in time $O(nT_{\mathrm{enc}}(n) + n^2 \log n)$.
- Each of the iterations of Step 3 can be computed in time $O(T_{\mathrm{dec}}(N) \log n + \ell n \log n)$ (recall that we repeat the list decoding algorithm for $O(\log n)$ times).

Therefore, the row-decoding phase runs in time $\widetilde{O}(\ell N^2)$.

Putting all together. The algorithm M runs first the column-decoding phase and then runs the row-decoding phase for $\widetilde{F} = (\widetilde{D})^\top$. The matrix $\widetilde{D}$ obtained by the column-decoding phase agrees with $D := ABQ^\top$ in at least $(\alpha/2) \cdot N$ columns with probability $1 - \ell/N^2$. Conditioned on this event, the row-decoding phase outputs AB with probability $1 - \ell/n^2$. Therefore, M outputs AB with probability $(1 - \ell/N^2)(1 - \ell/n^2) = 1 - o(1)$.

The total running time is at most $\ell \cdot \widetilde{O}(N^2)$. □

4.4 Worst-Case-to-Average-Case Reduction

In this section, we present a worst-case-to-average-case reduction for approximate matrix multiplication based on a left-right encoding with respect to a list-decodable code having a full-rank partition. To this end, we need two auxiliary results. The first one is a worst-case-to-average-case reduction for matrix multiplication.

Lemma 4.23 ([HS23, Theorem 1.6])**.** *Let $\mathbb{F}_p$ be an arbitrary finite field. There is a $\mathsf{polylog}(1/\alpha)/\alpha \cdot \widetilde{O}(n^2)$-time oracle algorithm $M^{\mathcal{O}}$ such that for any randomized oracle $\mathcal{O}$ such that for every large n,*

$$\Pr_{\substack{A,B \sim \mathbb{F}_p^{n \times n} \\ \mathcal{O}}} [\mathcal{O}(A, B) = AB] \ge \alpha,$$

it holds that for all large n and all $A, B \in \mathbb{F}_p^{n \times n}$,

$$\Pr_{M^{\mathcal{O}}} \left[M^{\mathcal{O}}(A, B) = AB \right] \ge \frac{2}{3}.$$

Moreover, $M^{\mathcal{O}}$ makes at most $(\mathsf{polylog}(1/\alpha)/\alpha) \cdot O(\log n)$ queries to the oracle $\mathcal{O}$.

Next, we reduce multiplying random $m \times n$ and $n \times m$ matrices to multiplying two random $n \times n$ matrices for $m \le n$.

Lemma 4.24. *There exists an $O(n^2)$-time one-query oracle algorithm $M^{\mathcal{O}}$ such that for every randomized oracle $\mathcal{O}$ such that for every sufficiently large n,*

$$\mathop{\mathbb{E}}_{\substack{\overline{A},\overline{B}\sim\mathbb{F}_p^{n\times n}\\ \mathcal{O}}}\left[\mathrm{dist}(\mathcal{O}(\overline{A},\overline{B}),\overline{AB})\right]\le 1-\alpha,$$

then, for every sufficiently large n and $\ell,m\le n$, it holds that

$$\mathop{\mathbb{E}}_{\substack{A\sim\mathbb{F}_p^{\ell\times n}\\ B\sim\mathbb{F}_p^{n\times m}\\ M^{\mathcal{O}}}}\left[\mathrm{dist}\big(M^{\mathcal{O}}(A,B),AB\big)\right]\le 1-\alpha.$$

Proof. The algorithm $M^{\mathcal{O}}$ runs on input $(A,B)\in\mathbb{F}_p^{\ell\times n}\times\mathbb{F}_p^{n\times m}$ as follows:

1. Sample $\overline{A},\overline{B}\sim\mathbb{F}_p^{n\times n}$, $I\sim\binom{[n]}{\ell}$ and $J\sim\binom{[n]}{m}$.
2. Replace $\overline{A}|_{I,[n]}$ with A. Specifically, if $I=\{i_1,\dots,i_\ell\}$ for $i_1<\cdots<i_\ell$, then the i_a-th row of $\overline{A}$ is the a-th row of A for every $a\in[\ell]$. Similarly, replace $\overline{B}|_{[n],J}$ with B.
3. Output $\mathcal{O}(\overline{A},\overline{B})|_{I,J}$.

Note that $M^{\mathcal{O}}$ runs in time $O(n^2)$. We prove the correctness. By calculation, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\substack{A\sim\mathbb{F}_p^{\ell\times n}\\ B\sim\mathbb{F}_p^{n\times m}\\ M^{\mathcal{O}}}}\left[\mathrm{dist}\big(M^{\mathcal{O}}(A,B),AB\big)\right] &= \mathop{\mathbb{E}}_{\substack{A\sim\mathbb{F}_p^{\ell\times n}\\ B\sim\mathbb{F}_p^{n\times m}\\ \overline{A},\overline{B}\sim\mathbb{F}_p^{n\times n}\\ I\sim\binom{[n]}{\ell},J\sim\binom{[n]}{m}}}\left[\mathrm{dist}\big(\mathcal{O}(\overline{A},\overline{B})|_{I,J},(\overline{AB})|_{I,J}\big)\,\middle|\,\overline{A}|_{I,[n]}=A,\overline{B}|_{[n],J}=B\right]\\
&= \mathop{\mathbb{E}}_{\substack{\overline{A},\overline{B}\sim\mathbb{F}_p^{n\times n}\\ I,J}}\left[\mathrm{dist}\big(\mathcal{O}(\overline{A},\overline{B})|_{I,J},(\overline{AB})|_{I,J}\big)\right]\\
&= \mathop{\Pr}_{\substack{\overline{A},\overline{B}\sim\mathbb{F}_p^{n\times n}\\ I,J\\ i\sim I,j\sim J}}\left[\mathcal{O}(\overline{A},\overline{B})_{i,j}\neq(\overline{AB})_{i,j}\right]\\
&= \mathop{\mathbb{E}}_{\overline{A},\overline{B}\sim\mathbb{F}_p^{n\times n}}\left[\mathrm{dist}(\mathcal{O}(\overline{A},\overline{B}),\overline{AB})\right]\\
&\le 1-\alpha.
\end{aligned}$$

Therefore, $M^{\mathcal{O}}$ satisfies the claim. □

We prove a worst-case-to-average and exact-to-approximate reduction for matrix multiplication.

Theorem 4.25. *Let $c,\alpha>0$ be parameters. Let $\mathsf{Enc}\colon\mathbb{F}_p^n\to\mathbb{F}_p^N$ be an encoding that can be computed in time $\widetilde{O}(n)$ and is ℓ-list decodable within radius $1-\frac{(1-c)\alpha}{2}$ for $\ell=O_{c,\alpha}(1)$ associated with a list decoding algorithm running in time $\ell\cdot\widetilde{O}(N)$. Moreover, suppose that Enc has an (m,δ)-full-rank partition that can be computed in time $\widetilde{O}(Nn)$ for $\delta\le c\alpha/6$.*

Then, there exists a randomized oracle algorithm M such that for every randomized oracle $\mathcal{O}$ such that

$$\mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ \mathcal{O}}}[\mathrm{dist}(\mathcal{O}(A,B),AB)] \le 1-\alpha, \tag{10}$$

it holds that for every $A,B\in\mathbb{F}_p^{n\times n}$,

$$\Pr_{M^{\mathcal{O}}}\left[M^{\mathcal{O}}(A,B)=AB\right] \ge \frac{2}{3}.$$

Moreover, the algorithm $M^{\mathcal{O}}$ makes $(N/m)^2\cdot O\left(\frac{1}{c\alpha}\right)\cdot O(\log n)$ queries to the oracle $\mathcal{O}$ and runs in time $\mathsf{poly}\left(\frac{1}{c\alpha}\right)\cdot\ell\cdot\widetilde{O}(N^2)$.

To prove Theorem 4.25, we invoke the well-known randomized $O(n^2)$-time verification algorithm for matrix multiplication.

Lemma 4.26 (Freivalds [Fre79])**.** *There exists a randomized $O(n^2)$-time algorithm M that is given $A,B,C\in\mathbb{F}_p^{n\times n}$ as input and satisfies the following:*

- *If $AB=C$, the algorithm $M(A,B,C)$ outputs Yes with probability 1.*
- *If $AB\neq C$, the algorithm $M(A,B,C)$ outputs No with probability $2/3$.*

Here, the probability is over the internal randomness of M.

Proof of Theorem 4.25. Let $\mathsf{Enc}\colon\mathbb{F}_p^n\to\mathbb{F}_p^N$ be the encoding of Theorem 4.25 associated with an (m,δ)-full-rank partition $\mathcal{P}=P_1\sqcup\cdots\sqcup P_a\sqcup P'$. For simplicity in explanation, write $P_{a+1}=P'$. Write $\mathsf{Enc}\colon x\mapsto Qx$ for a matrix $Q\in\mathbb{F}_p^{N\times n}$. Fix a randomized oracle $\mathcal{O}$ that satisfies Eq. (10).

For $A,B\in\mathbb{F}_p^{n\times n}$, consider $QA\in\mathbb{F}_p^{N\times n}$ and $BQ^\top\in\mathbb{F}_p^{n\times N}$. For every pair $P_i,P_j\in\mathcal{P}$ of subsets in the partition for $i,j\in[a]$, consider the submatrices $Q|_{P_i}\cdot A\in\mathbb{F}_p^{m\times n}$ and $B\cdot(Q|_{P_j})^\top\in\mathbb{F}_p^{n\times m}$. Since $\mathcal{P}$ is an (m,δ)-full-rank partition, the matrices $Q|_{P_i}$ and $Q|_{P_j}$ are full rank. From Lemma 4.5, if $A,B\sim\mathbb{F}_p^{n\times n}$ are chosen uniformly at random, then the marginal distribution of the pair $\left(Q|_{P_i}A,B(Q|_{P_j})^\top\right)$ is uniform over $\mathbb{F}_p^{m\times n}\times\mathbb{F}_p^{n\times m}$. Thus, from Lemma 4.24 and Eq. (10), there exists a one-query randomized algorithm $M_0^{\mathcal{O}}$ such that, for every $i,j\in[a]$, it holds that

$$\mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M_0^{\mathcal{O}}}}\left[\mathrm{dist}\left(M_0^{\mathcal{O}}\left(Q|_{P_i}A,B(Q|_{P_j})^\top\right),\,Q|_{P_i}A\cdot B(Q|_{P_j})^\top\right)\right]\le 1-\alpha. \tag{11}$$

Let $\mathsf{Enc}^\star\colon\mathbb{F}_p^{n\times n}\to\mathbb{F}_p^{N\times N}$ be the left-right encoding with respect to Enc. Then, the matrix $\mathsf{Enc}^\star(AB)=QABQ^\top$ can be obtained by aligning $(a+1)^2$ submatrices $Q|_{P_i}\cdot A\cdot B\cdot(Q|_{P_j})^\top\in\mathbb{F}_p^{m\times m}$ for all $i,j\in[a+1]$ (see Fig. 3). The number of entries of $\mathsf{Enc}^\star(AB)$ corresponding to the rest part $P_{a+1}=P'$ is at most $2\delta N^2\le(c\alpha/3)N^2$.

With this in mind, consider the oracle algorithm $M_1^{\mathcal{O}}$ that on input $A,B\in\mathbb{F}_p^{n\times n}$ runs as follows:

1. For each $i,j\in[a]$, compute $\widetilde{C}_{i,j}:=M_0^{\mathcal{O}}(Q|_{P_i}A,\,B(Q|_{P_j})^\top)$.
2. Let $\widetilde{C}_{a+1,a+1}\in\mathbb{F}_p^{|P_{a+1}|\times|P_{a+1}|}$ be an arbitrary matrix. Similarly, for $i\in[a]$, let $\widetilde{C}_{i,a+1}\in\mathbb{F}_p^{m\times|P_{a+1}|}$ and $\widetilde{C}_{a+1,j}\in\mathbb{F}_p^{|P_{a+1}|\times m}$ be arbitrary matrices.

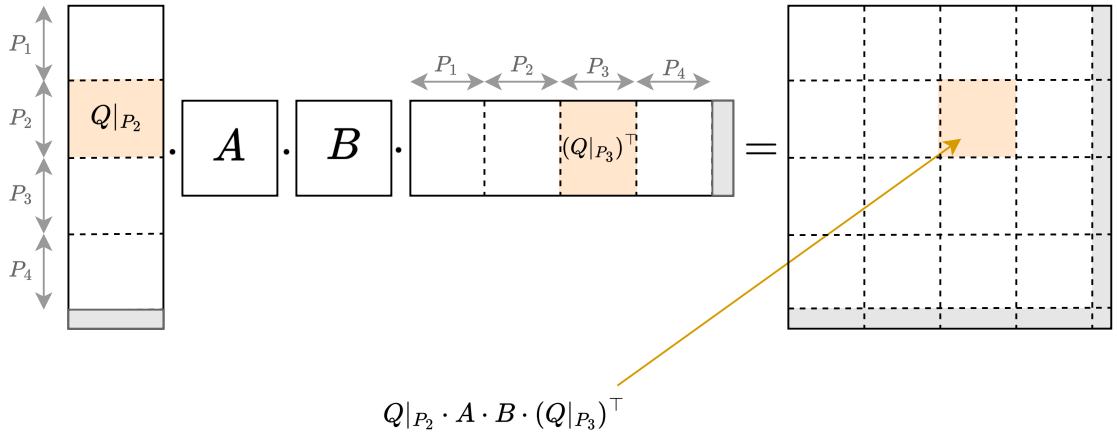

Figure 3: Each block of $\mathsf{Enc}^\star(AB)$ is equal to $Q|_{P_i} \cdot A \cdot B \cdot Q|_{P_j}^\top \in \mathbb{F}_p^{m \times m}$. If $A, B \sim \mathbb{F}_p^{n \times n}$, then each block is the product of two uniformly random matrices. The number of entries of $\mathsf{Enc}^\star(AB)$ whose index is in P' (the gray area in the right matrix) is at most $2\delta N^2 \le (c\alpha/3)N^2$.

3. Output the matrix $\widetilde{C} \in \mathbb{F}_p^{N \times N}$ obtained by aligning $\widetilde{C}_{i,j}$ for all $i, j \in [a+1]$.

Since $\delta \le \frac{c\alpha}{6}$, from Eq. (11), we have

$$\begin{aligned}\mathop{\mathbb{E}}_{\substack{A,B \sim \mathbb{F}_p^{n \times n} \\ M_1^{\mathcal{O}}}} \left[\mathrm{dist}(M_1^{\mathcal{O}}(A,B), \mathsf{Enc}^\star(AB))\right] &\le \mathop{\mathbb{E}}_{i,j \sim [a]} \left[\mathop{\mathbb{E}}_{\substack{A,B \sim \mathbb{F}_p^{n \times n} \\ M_0^{\mathcal{O}}}} \left[\mathrm{dist}\left(\widetilde{C}_{i,j},\, Q|_{P_i} \cdot A \cdot B \cdot (Q|_{P_j})^\top\right)\right] \right] + 2\delta \\ &\le 1 - \left(1 - \frac{c}{3}\right)\alpha.\end{aligned}$$

We say that an instance $(A, B) \in (\mathbb{F}_p^{n \times n})^2$ is *good* if

$$\mathop{\mathbb{E}}_{M_1^{\mathcal{O}}} \left[\mathrm{dist}(M_1^{\mathcal{O}}(A,B), \mathsf{Enc}^\star(AB))\right] \le 1 - \left(1 - \frac{c}{3}\right)^2 \alpha.$$

By Markov's inequality (Lemma 4.1), at least $\frac{c(1-c/3)}{3} \cdot \alpha$-fraction of $(A, B) \sim (\mathbb{F}_p^{n \times n})^2$ are good.

Again, by Markov's inequality, for every good instance (A, B), we have

$$\Pr_{M_1^{\mathcal{O}}} \left[\mathrm{dist}(M_1^{\mathcal{O}}(A,B), \mathsf{Enc}^\star(AB)) \le 1 - \left(1 - \frac{c}{3}\right)^3 \alpha\right] \ge \frac{c}{3} \cdot \left(1 - \frac{c}{3}\right)^2 \alpha = \Omega(c\alpha). \tag{12}$$

Let $M_2^{\mathcal{O}}$ be the algorithm that runs on input (A, B) as follows:

1. Compute $\widetilde{C} = M_1^{\mathcal{O}}(A, B)$ and run the algorithm M of Lemma 4.21 on input $A, B, \widetilde{C}$.
2. If M outputs AB (this can be checked by Lemma 4.26 in time $O(n^2)$), output it.

3. Repeat Steps 1 and 2 for $O(\frac{1}{c\alpha})$ times. If the algorithm outputs noting during the iteration, output $\perp$.

From Eq. (12), if the input (A, B) is good, then $\operatorname{dist}(\widetilde{C}, \mathsf{Enc}^\star(AB)) \le 1-(1-c/3)^3\alpha \le 1-(1-c)\alpha$ with probability $\Omega(c\alpha)$ over the internal randomness of $M_1^{\mathcal{O}}$. Conditioned on this event, from Lemma 4.21, the algorithm M outputs AB at Step 2 with probability $2/3$ (over the internal randomness of M) since Enc is ℓ-list decodable within radius $1-(1-c)\alpha/2$. Therefore, $M_2^{\mathcal{O}}(A, B)$ outputs AB with probability $2/3$ for any good (A, B); thus

$$\Pr_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M_2^{\mathcal{O}}}}\left[M_2^{\mathcal{O}}(A,B) = AB\right] \ge \Omega(c\alpha).$$

Finally, from Lemma 4.23, we obtain an oracle algorithm that computes AB with probability $2/3$ for any input $A, B \in \mathbb{F}_p^{n\times n}$. Since M_2 runs in time $\ell \cdot \widetilde{O}(N^2)$, the algorithm M_3 runs in time $\ell \cdot \mathsf{poly}(\frac{1}{c\alpha}) \cdot \widetilde{O}(N^2)$. Consider the number of queries. Since $M_2^{\mathcal{O}}$ makes $a^2 \cdot O(\frac{1}{c\alpha}) \le (N/m)^2 \cdot O(\frac{1}{c\alpha})$ queries, the algorithm M_3 makes $(N/m)^2 \cdot O(\frac{1}{c\alpha}) \cdot O(\log n)$ queries. □

4.5 Large Field via Reed–Solomon Codes

We present an exact-to-approximate reduction for matrix multiplication over large $\mathbb{F}_p$, proving Theorem 1.3. This follows from Theorem 4.25 using the Reed–Solomon codes as Enc.

It is known that Reed–Solomon codes are nearly linear time encodable and list-decodable. For arbitrary field $\mathbb{F}$, let $\mathrm{mult}(n)$ be the time needed for multiplying two univariate polynomials over $\mathbb{F}$ of degree n. It is known that $\mathrm{mult}(n) = O(n \log n)$ for any finite field $\mathbb{F} = \mathbb{F}_p$ (e.g., [GG13, Corollary 8.19]).

Lemma 4.27 (Fast Multipoint Evaluation for Univariate Polynomials [Fid72])**.** *There exists an algorithm that, given N distinct points $\gamma_1, \dots, \gamma_N \in \mathbb{F}_p$ and coefficients of a univariate polynomial p of degree at most $N-1$ over $\mathbb{F}_p$, outputs $(p(\gamma_1), \dots, p(\gamma_N))$ using $O(\mathrm{mult}(N) \log N)$ arithmetic operations of $\mathbb{F}_p$.*

In particular, for any $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_N)$, we can compute $\mathsf{RS}_{p,n,\boldsymbol{\gamma}}$ in time $\widetilde{O}(N)$.

Lemma 4.28 (List-Decoding for Reed–Solomon Codes; Alekhnovich [Ale05, Corollary 1.3])**.** *For $n, N \in \mathbb{N}$, a finite field $\mathbb{F}_p$ and any $\alpha > 0$ satisfying $\alpha > \sqrt{n/N}$, for any $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_N)$, the Reed–Solomon encoding $\mathsf{RS}_{p,n,\boldsymbol{\gamma}} \colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ is ℓ-list decodable within radius $1-\alpha$ by an $(N/n)^{O(1)} \cdot \mathrm{mult}(N) \log N$-time list decoding algorithm for $\ell = O(1/\alpha)$.*[9]

In particular, for any given $n \in \mathbb{N}$ and $\alpha > 0$, for $N \ge \alpha^{-2} n$ and any sequence $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_N)$ of distinct elements of $\mathbb{F}_p$ with $p \ge N$, we can list-decode $\mathsf{RS}_{p,n,\boldsymbol{\gamma}}$ in time $\mathsf{poly}(1/\alpha) \cdot \widetilde{O}(N)$.

Proof of Theorem 1.3. We apply Theorem 4.25 for $c = 1/2$ and Enc being a Reed–Solomon encoding $\mathsf{RS}_{p,n,\boldsymbol{\gamma}}$, where $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_N)$ are N distinct points of $\mathbb{F}_p$ for $N = \lceil \alpha^{-2} \rceil \cdot n$ and $p > N$ is a prime power. From Lemmas 4.27 and 4.28, $\mathsf{RS}_{p,n,\boldsymbol{\gamma}}$ can be computed in time $\widetilde{O}(n)$ and is ℓ-list decodable in time $\mathsf{poly}(1/\alpha) \cdot \widetilde{O}(N)$ for $\ell = O(1/\alpha)$. Moreover, from Lemma 4.8, Enc has an $(n, 0)$-full-rank partition.

[9] The bound on the list size directly follows from the Johnson bound [Joh62].

Let $\mathcal{O}$ be any oracle algorithm satisfying the condition of Theorem 1.3. The oracle algorithm $M^{\mathcal{O}}$ of Theorem 4.25 computes AB for any $A, B \in \mathbb{F}_p^{n\times n}$ with probability $2/3$ (over the internal randomness of $M^{\mathcal{O}}$). The running time of M is $\mathsf{poly}(1/\alpha)\cdot\widetilde{O}(n^2)$ and M makes at most $(N/m)^2\cdot\mathsf{poly}(1/\alpha)\cdot O(\log n) = \mathsf{poly}(1/\alpha)\cdot O(\log n)$ queries. □

4.6 Small Field via Walk-Amplified Codes

We present an exact-to-approximate reduction for matrix multiplication over small $\mathbb{F}_p$ using the walk-amplified encoding (Definition 4.13), which is list decodable in nearly linear time [Jer23; JST21].

Lemma 4.29 (Special case of [Jer23, Theorem 7.5 of the full version])**.** *Let $\mathbb{F}_p$ be any prime field. Let $\mathsf{Enc} = \mathsf{Enc}_1 \circ \mathsf{Enc}_0$ be a (k, d, λ, k)-walk-amplified encoding with base encoding $\mathsf{Enc}_0\colon \mathbb{F}_p^n \to \mathbb{F}_p^{n'}$ and let $\beta > 0$ be any small parameter. Suppose that Enc_0 is uniquely decodable within radius ρ_0 by an $\widetilde{O}(n')$-time decoding algorithm and $k, \lambda, \rho_0, \beta$ satisfy*

$$\beta \geq \max\left\{(2^{20}\cdot\lambda\cdot k^3)^{1/2}, 4(1-(C_p\rho_0/4)^2)^{k/2}\right\}, \tag{13}$$

where $C_p := 1-\cos(\pi/p)$. Then, the walk-amplified code Enc is ℓ-list-decodable within radius $(1-1/p)(1-\beta)$ and there exists an ℓ-list decoding randomized algorithm that runs in time $2^{p^{O(k^3/\beta^2)}}\cdot\widetilde{O}(N)$.

As the base encoding Enc_0, we use the following explicit family of linear-time encodable and unique-decodable encoding.

Lemma 4.30 ([Jer23, Theorem 7.16 of the full version])**.** *Let $\mathbb{F}_p$ be any finite field. There exists universal constants $c, \rho_0 > 0$ such that, for every n, there exists $n \leq n' \leq cn$ and an encoding $\mathsf{Enc}_0\colon \mathbb{F}_p^n \to \mathbb{F}_p^{n'}$ that can be computed in time $O(n)$ and is unique-decodable within radius ρ_0 associated with an $O(n')$-time unique-decoding algorithm.*

Proof of Theorem 1.4. Let $\mathcal{O}$ be the oracle that satisfies

$$\mathop{\mathbb{E}}_{A,B\sim\mathbb{F}_p^{n\times n}}[\mathrm{dist}(\mathcal{O}(A,B), AB)] \leq 1-\frac{2}{p}-\varepsilon.$$

We apply Theorem 4.25 for $\alpha = \frac{2}{p}+\varepsilon$ and $c = \frac{p\varepsilon}{6}$. Set $\beta = \frac{\varepsilon}{4}$ and $\delta = \frac{c\alpha}{6} \leq \varepsilon$. As the encoding Enc, we use the (k, d, λ, k)-walk-amplified encoding of Definition 4.13 using the encoding Enc_0 of Lemma 4.30 as the base encoding. The expander graph used by the direct sum encoding is constructed as follows: Choose $k \geq 2, \lambda > 0$ so that Eq. (13) and the condition $(2\lambda)^k \leq \frac{\delta^2}{288}$ of Lemma 4.16 hold. In particular, we can set $k = Cp^4\log(1/\varepsilon)$ and $\lambda = \frac{\varepsilon^2}{Ck^3}$ for a sufficiently large constant $C > 2^{20}$. Then, choose $m = \Theta(n)$ and $d = O(1)$ according to Lemma 4.16. For $d' = \max\{d, 4/\lambda^2\}$, use a random d'-regular graph as the expander graph in the direct sum encoding. From Lemma 4.12, the random d-regular graph is a $2/\sqrt{d}$-expander with probability $1-n^{-10}$ and has girth at least k with probability $\Omega(1)$. Moreover, this graph can be constructed in time $O(n)$.

From Lemma 4.29, the (k, d, λ, k)-walk-amplified encoding described above can be computed in time $O_{p,\varepsilon}(n)$ and ℓ-list decodable within radius

$$\left(1-\frac{1}{p}\right)(1-\beta) \geq 1-\frac{1}{p}-\beta \geq 1-\frac{1}{p}-\frac{(1-c)\alpha}{2}$$

in time $\widetilde{O}_{p,\varepsilon}(N)$. Moreover, from Lemma 4.14, this encoding has an (m, δ)-full-rank partition for $m = \lfloor \frac{\lambda n}{2k} \rfloor$.

Finally, from Theorem 4.25, there exists a randomized oracle algorithm M that computes AB for any given A, B with probability $2/3$ in time $\widetilde{O}_{p,\varepsilon}(N^2)$. The number of queries to the oracle $\mathcal{O}$ made by $M^{\mathcal{O}}$ is at most $(N/m)^2 \cdot O\left(\frac{1}{c\alpha}\right) \cdot O(\log n) \leq 2^{\mathsf{poly}(p,1/\varepsilon)} \cdot O(\log n)$. □

5 Online Matrix-Vector Multiplication

In the online matrix-vector multiplication (OMv), we are given an $n \times n$ matrix A, which is to be preprocessed in polynomial time. Then, we are asked to compute Av efficiently (faster than the trivial $O(n^2)$ time) for n-dimensional vectors v given online. We focus on OMv over a finite field $\mathbb{F}_p$. Throughout this section, the computational time of an algorithm amounts to the number of arithmetic operation over $\mathbb{F}_p$.

To state it more formally, we borrow the notion of a data-structure algorithm from [HS23, Section 6 of the full version].

Definition 5.1 (Data Structure Algorithm)**.** *A (randomized)* data-structure algorithm M *consists of a pair of (randomized) algorithms* $(M_{\mathrm{pre}}, M_{\mathrm{query}})$.

- *The* preprocess M_{pre} *is a polynomial-time algorithm that is given an input x and outputs a string $\pi = M_{\mathrm{pre}}(x)$ called a* data structure.
- *The* query algorithm M_{query} *is given an input v as query and oracle access to the data structure π and then outputs $M^{\pi}_{\mathrm{query}}(v)$.*

We call an input to M_{pre} preprocess input *and input to M_{query}* query input*. Let $M(A; v) = M^{\pi}_{\mathrm{query}}(v)$ for $\pi = M_{\mathrm{pre}}(A)$ denote the final output of M on query v. The* preprocess time *and* query time *are the running times of M_{pre} and M_{query}, respectively. For a data-structure oracle $\mathcal{O} = (\mathcal{O}_{\mathrm{pre}}, \mathcal{O}_{\mathrm{query}})$ and a pair of oracle algorithms $M = (M_{\mathrm{pre}}, M_{\mathrm{query}})$, define $M^{\mathcal{O}} = \left(M^{\mathcal{O}_{\mathrm{pre}}}_{\mathrm{pre}}, M^{\mathcal{O}_{\mathrm{query}}}_{\mathrm{query}}\right)$.*

In the average-case version of OMv, the preprocess input and query input are chosen uniformly at random. In the following, we reduce the exact version of OMv on a worst-case input to the approximate version of OMv on a random instance.

Theorem 5.2. *Let $n \in \mathbb{N}$ and $\alpha = \alpha(n) > 0$ be parameters and $\mathbb{F}_p$ be a finite field of order $p = p(n)$ such that $p > n/\alpha^2$. Suppose there exists a randomized data-structure oracle $\mathcal{O} = (\mathcal{O}_{\mathrm{pre}}, \mathcal{O}_{\mathrm{query}})$ such that for all sufficiently large $n \in \mathbb{N}$,*

$$\mathop{\mathbb{E}}_{\substack{A \sim \mathbb{F}_p^{n \times n} \\ v \sim \mathbb{F}_p^n \\ \mathcal{O}}} [\mathrm{dist}(\mathcal{O}(A; v),\ Av)] \leq 1 - \alpha.$$

Then, there exists a randomized $O(1/\alpha^3)$-query oracle algorithm $M^{\mathcal{O}}$ that runs in time $\mathsf{poly}(1/\alpha) \cdot \widetilde{O}(n^2)$ and satisfies, for all large $n \in \mathbb{N}$ and all $A, B \in \mathbb{F}_p^{n \times n}$,

$$\Pr_{M^{\mathcal{O}}}\left[M^{\mathcal{O}}(A; v) = Av\right] \geq \frac{2}{3}.$$

Here, we assume that the arithmetic operation of $\mathbb{F}_p$ can be performed in constant time.

Theorem 5.3 (full version of Theorem 1.5)**.** *Let $n \in \mathbb{N}$ and $\varepsilon = \varepsilon(n) > 0$ be parameters and $\mathbb{F}_p$ be a finite field of prime order $p = p(n)$. Suppose that there exists a randomized data-structure oracle $\mathcal{O} = (\mathcal{O}_{\mathrm{pre}}, \mathcal{O}_{\mathrm{query}})$ such that for all sufficiently large $n \in \mathbb{N}$,*

$$\mathop{\mathbb{E}}_{\substack{A \sim \mathbb{F}_p^{n \times n} \\ v \sim \mathbb{F}_p^n \\ \mathcal{O}}} [\mathrm{dist}(\mathcal{O}(A; v),\ Av)] \leq 1 - \frac{1}{p} - \varepsilon.$$

Then, there exists a randomized $2^{\mathsf{poly}(p, 1/\varepsilon)} \cdot O(\log n)$-query $2^{p^{\mathsf{poly}(p, 1/\varepsilon)}} \cdot \widetilde{O}(n)$-time oracle data-structure algorithm $M^{\mathcal{O}}$ such that for all large n and all $A \in \mathbb{F}_p^{n \times n}, v \in \mathbb{F}_p^n$,

$$\Pr_{M^{\mathcal{O}}} \left[M^{\mathcal{O}}(A; v) = Av \right] \geq \frac{2}{3}.$$

Here, we assume that the arithmetic operation of $\mathbb{F}_p$ can be performed in constant time.

To prove Theorems 5.2 and 5.3, we present the following general exact-to-approximate and worst-case-to-average-case reduction for OMv.

Theorem 5.4 (Worst-Case-to-Average-Case Reduction)**.** *Let $c, \alpha > 0$ be parameters and $\mathbb{F}_p$ be a finite field. Let $\mathsf{Enc} \colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ be an encoding that can be computed in time $\widetilde{O}(n)$ and is ℓ-list-decodable within radius $1 - (1 - c)\alpha$ for $\ell = O_{c,\alpha}(1)$ by a list-decoding algorithm running in time $\ell \cdot \widetilde{O}(N)$. Suppose that Enc has an (m, δ)-full-rank partition that can be computed in polynomial time for $\delta \leq \frac{c\alpha}{3}$. Then, there exists a one-query randomized oracle data-structure algorithm $M = (M_{\mathrm{pre}}, M_{\mathrm{query}})$ such that, for any randomized data-structure oracle $\mathcal{O} = (\mathcal{O}_{\mathrm{pre}}, \mathcal{O}_{\mathrm{query}})$ that satisfies*

$$\mathop{\mathbb{E}}_{\substack{A \sim \mathbb{F}_p^{n \times n}, v \sim \mathbb{F}_p^n \\ \mathcal{O}}} [\mathrm{dist}(\mathcal{O}(A; v), Av)] \leq 1 - \alpha,$$

it holds that for every $A \in \mathbb{F}_p^{n \times n}$ and $v \in \mathbb{F}_p^n$,

$$\Pr_{M^{\mathcal{O}}} \left[M^{\mathcal{O}}(A; v) = Av \right] \geq \frac{2}{3}.$$

Moreover, each $M_{\mathrm{pre}}, M_{\mathrm{query}}$ makes at most $(N/m) \cdot O\left(\frac{\log n}{c\alpha}\right)$ oracle calls and M_{query} runs in time $\mathsf{poly}\left(\frac{1}{c\alpha}\right) \cdot \ell \cdot \widetilde{O}(N)$.

To prove Theorem 5.4, we need some auxiliary results. First, we borrow the following efficient verification data-structure algorithm for OMv.

Lemma 5.5 ([HS23, Lemma 6.2 of the full version])**.** *For any $d \in \mathbb{N}$, there exists a randomized data-structure algorithm $M = (M_{\mathrm{pre}}, M_{\mathrm{query}})$ such that, given any matrix $A \in \mathbb{F}_p^{m \times n}$ as preprocess input and any vectors $v \in \mathbb{F}_p^n, w \in \mathbb{F}_p^m$ as query input, $M(A; v, w)$ decides in query time $O(dn)$ whether $Av = w$ correctly with probability $1 - 2^{-d}$ (over the internal randomness of M).*

Second, by combining a list-decodable code with the data-structure verification algorithm, we present the following exact-to-approximate reduction.

Lemma 5.6. *Let $\mathbb{F}_p$ be a finite field and $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ be an encoding represented as $\mathsf{Enc}\colon x \mapsto Qx$ that can be computed in time $\widetilde{O}(n)$ and is ℓ-list decodable within radius $1-\alpha$ by an $\widetilde{O}(N)$-time list decoding algorithm for $\ell = O_{r,p}(1)$. Then, there exists a randomized data-structure algorithm $M = (M_{\mathrm{pre}}, M_{\mathrm{query}})$ such that, for any $A \in \mathbb{F}_p^{n\times n}$, $v \in \mathbb{F}_p^n$ and $\tilde{w} \in \mathbb{F}_p^N$ such that $\mathrm{dist}(\tilde{w}, QA\cdot v) \le 1-\alpha$, $M(A; v, \tilde{w})$ outputs Av with probability $2/3$ (over the internal randomness of M). Moreover, M_{query} runs in time $\widetilde{O}(N)$.*

Proof. In the preprocess phase, run the preprocess algorithm of Lemma 5.5. In the query phase, given $v \in \mathbb{F}_p^n$ and $\tilde{w} \in \mathbb{F}_p^N$ as input, run the list-decoding algorithm of Enc on input $\tilde{w}$. If $\tilde{w}$ is $(1-\alpha)$-close to $Av \in \mathbb{F}_p^n$, then we obtain a list of ℓ vectors that contains Av. We can identify Av from the list by the data-structure algorithm of Lemma 5.5. □

Third, we reduce multiplying a random $m \times n$ matrix and an n-dimensional random vector to multiplying an $n \times n$ random matrix and an n-dimensional random vector.

Lemma 5.7. *There exists a one-query oracle data-structure algorithm $M^{\mathcal{O}}$ such that for every randomized data-structure oracle $\mathcal{O} = (\mathcal{O}_{\mathrm{pre}}, \mathcal{O}_{\mathrm{query}})$ such that for every large n,*

$$\mathop{\mathbb{E}}_{\substack{\bar{A}\sim\mathbb{F}_p^{n\times n},\\ v\sim\mathbb{F}_p^n,\\ \mathcal{O}}} \left[\mathrm{dist}\big(\mathcal{O}(\bar{A}; v), \bar{A}v\big)\right] \le 1-\alpha,$$

then, for all large n and $m \le n$,

$$\mathop{\mathbb{E}}_{\substack{A\sim\mathbb{F}_p^{m\times n},\\ v\sim\mathbb{F}_p^n,\\ M^{\mathcal{O}}}} \left[\mathrm{dist}\big(M^{\mathcal{O}}(A; v), Av\big)\right] \le 1-\alpha.$$

Proof. For the oracle $\mathcal{O} = (\mathcal{O}_{\mathrm{pre}}, \mathcal{O}_{\mathrm{query}})$, the oracle data-structure algorithm $M^{\mathcal{O}}$ runs as follows:

1. In the preprocess phase, on preprocess input $A \in \mathbb{F}_p^{m\times n}$,
   (a) Sample $\bar{A} \sim \mathbb{F}_p^{n\times n}$ and $I \sim \binom{[n]}{m}$.
   (b) Replace $\bar{A}|_{I,[n]}$ with A. Formally, if $I = \{i_1, \ldots, i_m\}$ for $i_1 < \cdots < i_m$, then the i_a-th row of $\bar{A}$ is replaced by the a-th row of A for every $a \in [m]$.
   (c) Run the preprocess oracle $\mathcal{O}_{\mathrm{pre}}$ on input $\bar{A}$.
2. In the query phase, on query input $v \in \mathbb{F}_p^n$, output $\mathcal{O}(\bar{A}; v)|_I$, where $I \subseteq [n]$ is the size-m subset of Step 1(a).

We prove the correctness of $M^{\mathcal{O}}$. By calculation, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\substack{A\sim\mathbb{F}_p^{m\times n},\\ v\sim\mathbb{F}_p^n,\\ M^{\mathcal{O}}}}\left[\mathrm{dist}\left(M^{\mathcal{O}}(A;v),Av\right)\right] &= \mathop{\mathbb{E}}_{\substack{A\sim\mathbb{F}_p^{m\times n}\\ v\sim\mathbb{F}_p^n\\ \bar{A}\sim\mathbb{F}_p^{n\times n}\\ I\sim\binom{[n]}{m}\\ \mathcal{O}}}\left[\mathrm{dist}\left(\mathcal{O}(\bar{A};v)|_I,\bar{A}v|_I\right)\mid \bar{A}|_I=A\right]\\
&= \mathop{\mathbb{E}}_{\substack{\bar{A}\sim\mathbb{F}_p^{n\times n}\\ v\sim\mathbb{F}_p^n\\ I\sim\binom{[n]}{m}\\ \mathcal{O}}}\left[\mathrm{dist}\left(\mathcal{O}(\bar{A};v)|_I,\bar{A}v|_I\right)\right]\\
&= \Pr_{\substack{\bar{A}\sim\mathbb{F}_p^{n\times n}\\ v\sim\mathbb{F}_p^n\\ I\sim\binom{[n]}{m}\\ i\sim I\\ \mathcal{O}}}\left[\mathcal{O}(\bar{A};v)_i\neq(\bar{A}v)_i\right]\\
&= \mathop{\mathbb{E}}_{\substack{\bar{A}\sim\mathbb{F}_p^{n\times n},\\ v\sim\mathbb{F}_p^n,\\ \mathcal{O}}}\left[\mathrm{dist}\left(\mathcal{O}(\bar{A};v),\bar{A}v\right)\right]\\
&\le 1-\alpha.
\end{aligned}$$

This proves the claim. □

Finally, we borrow the worst-case-to-average-case reduction for OMv.

Lemma 5.8 ([HS23, Theorem 6.1 of the full version])**.** *Let $\mathbb{F}_p$ be a finite field. There exists a randomized* $\mathsf{polylog}(1/\alpha)/\alpha\cdot\widetilde{O}(n^2)$*-time*[10] *oracle data-structure algorithm $M^{\mathcal{O}}$ such that for any randomized data-structure oracle $\mathcal{O}$ such that for every large n,*

$$\Pr_{\substack{A\sim\mathbb{F}_p^{n\times n}\\ v\sim\mathbb{F}_p^n\\ \mathcal{O}}}[\mathcal{O}(A;v)=Av]\ge\alpha,$$

it holds that for all large n and all $A\in\mathbb{F}_p^{n\times n}$ and $v\in\mathbb{F}_p^n$,

$$\Pr_{M^{\mathcal{O}}}\left[M^{\mathcal{O}}(A;v)=Av\right]\ge\frac{2}{3}.$$

Moreover, $M^{\mathcal{O}}$ makes at most $\mathsf{polylog}(1/\alpha)/\alpha\cdot O(\log n)$ *queries to $\mathcal{O}$.*

Proof of Theorem 5.4. Let $\mathsf{Enc}\colon\mathbb{F}_p^n\to\mathbb{F}_p^N$ be the encoding of Theorem 5.4 associated with an (m,δ)-full-rank partition $\mathcal{P}=P_1\sqcup\cdots\sqcup P_a\sqcup P'$. Write $\mathsf{Enc}\colon x\mapsto Qx$ for a matrix $Q\in\mathbb{F}_p^{N\times n}$.

For a random matrix $A\sim\mathbb{F}_p^{n\times n}$, consider $QA\in\mathbb{F}_p^{N\times n}$. For each $i\in[a]$, the submatrix $Q|_{P_i}$ that consists of row vectors of Q in P_i is column-full-rank; thus the marginal distribution of the submatrix $QA|_{P_i}=Q|_{P_i}\cdot A$ is uniform over $\mathbb{F}_p^{m\times n}$ from Lemma 4.5. From Lemma 5.7 and by the

[10] Here, we assume that the arithmetic operation of $\mathbb{F}_p$ can be performed in constant time.

assumption of $\mathcal{O}$, there exists a one-query randomized oracle data-structure algorithm $M_0^{\mathcal{O}}$ such that, for every $i \in [a]$, it holds that

$$\mathop{\mathbb{E}}_{\substack{A\sim\mathbb{F}_p^{n\times n}\\ v\sim\mathbb{F}_p^n\\ M_0^{\mathcal{O}}}}[\mathrm{dist}(M_0((QA)|_{P_i}; v),\ (QA)|_{P_i}\cdot v)] \le 1-\alpha. \tag{14}$$

For simplicity in explanation, write $P_{a+1} := P'$. Then, note that $QA \in \mathbb{F}_p^{N\times n}$ can be obtained by aligning $(QA)|_{P_i}$ for all $i \in [a+1]$. The number of entries of QAv whose row index is in P_{a+1} is at most $|P_{a+1}| \le \delta N$ since $\mathcal{P}$ is an (m,δ)-full-rank partition.

With this in mind, consider the following randomized oracle data-structure algorithm $M_1^{\mathcal{O}}$ that runs as follows:

1. In the preprocess phase, on preprocess input $A \in \mathbb{F}_p^{n\times n}$, compute $QA \in \mathbb{F}_p^{N\times n}$ and an (m,δ)-full-rank partition $\mathcal{P} = P_1 \sqcup \cdots \sqcup P_a \sqcup P'$. Finally, run the preprocess algorithm M_{pre} of Lemma 5.5. These can be done in $\mathsf{poly}(n)$ time.
2. In the query phase, on query input $v \in \mathbb{F}_p^n$,
   (a) For each $i \in [a]$, compute $\tilde{w}_i := M_0^{\mathcal{O}}((QA)|_{P_i}; v)$.
   (b) Let $\tilde{w}_{a+1} \in \mathbb{F}_p^{|P_{a+1}|}$ be an arbitrary vector.
   (c) Output the vector $\tilde{w} \in \mathbb{F}_p^N$ obtained by aligning $\tilde{w}_i$ for $i \in [a+1]$.

Since $\delta \le \frac{c\alpha}{3}$ and Eq. (14), we have

$$\begin{aligned}\mathop{\mathbb{E}}_{\substack{A\sim\mathbb{F}_p^{n\times n}\\ v\sim\mathbb{F}_p^n\\ M_1^{\mathcal{O}}}}\left[\mathrm{dist}\left(M_1^{\mathcal{O}}(A;v), QA\cdot v\right)\right] &\le \mathsf{E}_{i\sim[a]}\left[\mathop{\mathsf{E}}_{\substack{A\sim\mathbb{F}_p^{n\times n}\\ v\sim\mathbb{F}_p^n\\ M_1^{\mathcal{O}}}}[\mathrm{dist}(\tilde{w}_i, (QA)|_{P_i}\cdot v)]\right] + \delta \\ &\le 1-\left(1-\frac{c}{3}\right)\alpha.\end{aligned}$$

We say that an instance $(A,v) \sim \mathbb{F}_p^{n\times n}\times\mathbb{F}_p^n$ is *good* if

$$\mathsf{E}_{M_1^{\mathcal{O}}}\left[\mathrm{dist}\left(M_1^{\mathcal{O}}(A;v),\ QA\cdot v\right)\right] \le 1-\left(1-\frac{c}{3}\right)^2\alpha.$$

By Markov's inequality, at least $\frac{c(1-c/3)}{3}\cdot\alpha$-fraction of instances are good. Again, by Markov's inequality, for every good instance (A,v), we have

$$\Pr_{M_1^{\mathcal{O}}}\left[\mathrm{dist}\left(M_1^{\mathcal{O}}(A;v),\ QA\cdot v\right) \le 1-\left(1-\frac{c}{3}\right)^3\alpha\right] \ge \frac{c}{3}\left(1-\frac{c}{3}\right)^2\alpha = \Omega(c\alpha). \tag{15}$$

Let $M_2^{\mathcal{O}}$ be the data-structure algorithm that runs as follows:

1. Compute $\tilde{w} = M_1^{\mathcal{O}}(A;v)$ and compute $w := M(A;v,\tilde{w})$ where M is the data-structure algorithm of Lemma 5.6.

2. Check if $w = Av$ by the verification algorithm of Lemma 5.5. If $w = Av$, output w.

3. Repeat Steps 1 and 2 for $O\left(\frac{1}{c\alpha}\right)$ times. If $M_2^{\mathcal{O}}$ does not output Av during the iteration, output $\perp$.

We prove the correctness of $M_2^{\mathcal{O}}$. From Eq. (15), if the instance (A, v) is good, then we have $\mathrm{dist}(\tilde{w},\, QA \cdot v) \le 1-(1-c/3)^3\alpha \le 1-(1-c)\alpha$ with probability $\Omega(c\alpha)$ over the internal randomness of $M_1^{\mathcal{O}}$. Conditioned on this event, from Lemma 5.6, the algorithm M outputs Av at Step 2 with probability $2/3$ from Lemma 5.6 (note that Enc is ℓ-list-decodable within radius $1-(1-c)\alpha$). Therefore, $M_2^{\mathcal{O}}(A; v)$ outputs Av with probability $2/3$ for any good (A, v); thus

$$\Pr_{\substack{A\sim\mathbb{F}_p^{n\times n}\\ v\sim\mathbb{F}_p^N\\ M_2^{\mathcal{O}}}}\left[M_2^{\mathcal{O}}(A; v) = Av\right] \ge \Omega(c\alpha).$$

Finally, from Lemma 5.8, we obtain a randomized oracle data-structure algorithm M_3 that satisfies, for all large n and all $A \in \mathbb{F}_p^{n\times n}, v \in \mathbb{F}_p^n$,

$$\Pr_{M_3}\left[M_3^{\mathcal{O}}(A; v) = Av\right] \ge \frac{2}{3}.$$

□

5.1 Large Field via Reed–Solomon Codes

We present an exact-to-approximate reduction for OMv over large $\mathbb{F}_p$, proving Theorem 5.2. This can be obtained by using the Reed–Solomon codes in Theorem 5.4. Recall the Reed–Solomon encoding of Definition 4.7. From Lemmas 4.27 and 4.28, $\mathsf{RS}_{p,n,\boldsymbol{\gamma}}\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ can be computed in time $\widetilde{O}(n)$, and for any $\alpha > \sqrt{n/N}$, the encoding $\mathsf{RS}_{p,n,\boldsymbol{\gamma}}$ is $O(1/\alpha)$-list-decodable within radius $1-\alpha$ in time $\mathsf{poly}(1/\alpha)\cdot O(n)$.

Proof of Theorem 5.2. We apply Theorem 5.4 for $c = 1/2$ and Enc being a Reed–Solomon encoding $\mathsf{RS}_{p,n,\boldsymbol{\gamma}}$, where $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_N)$ are N distinct points of $\mathbb{F}_p$ for $N = \lceil \alpha^{-2} \rceil \cdot n$ and $p > N$ is a prime power. From Lemmas 4.27 and 4.28, $\mathsf{RS}_{p,n,\boldsymbol{\gamma}}$ can be computed in time $\tilde{O}(n)$ and is ℓ-list decodable in time $\mathsf{poly}(1/\alpha)\cdot\widetilde{O}(N)$ for $\ell = O(1/\alpha)$. Moreover, from Lemma 4.8, Enc has a uniform partition.

Let $\mathcal{O}$ be any data-structure oracle that satisfies the condition of Theorem 5.2. The oracle data-structure algorithm $M^{\mathcal{O}}$ of Theorem 5.4 computes Av for any $A \in \mathbb{F}_p^{n\times n}, v \in \mathbb{F}_p^n$ with probability $2/3$ (over the internal randomness of $M^{\mathcal{O}}$). The running time of $M^{\mathcal{O}}$ is $\mathsf{poly}(1/\alpha)\cdot\widetilde{O}(n)$ and M makes at most $(N/m)\cdot O(\log n/\alpha) = \mathsf{poly}(1/\alpha)\cdot O(\log n)$ queries. □

5.2 Small Field via Walk-Amplified Codes

We present an exact-to-approximate reduction for matrix multiplication over small $\mathbb{F}_p$ using the walk-amplified encoding (Definition 4.13), proving Theorem 5.3.

Proof of Theorem 5.3. Let $\mathcal{O}$ be the data-structure oracle that satisfies

$$\mathop{\mathbb{E}}_{\substack{A\sim\mathbb{F}_p^{n\times n}\\ v\sim\mathbb{F}_p^n\\ \mathcal{O}}}\left[\mathrm{dist}(\mathcal{O}(A; v),\, Av)\right] \le 1 - \frac{1}{p} - \varepsilon.$$

We apply Theorem 5.4 for $\alpha = \frac{1}{p} + \varepsilon$ and $c = \frac{p\varepsilon}{3}$. Set $\beta = \frac{2\varepsilon}{3}$ and $\delta = \frac{c\alpha}{3} \le \varepsilon$.

We use the (k, d, λ, k)-walk-amplified encoding considered in the proof of Theorem 1.4 presented in Section 4.6. As shown in Section 4.6, this (k, d, λ, k)-walk-amplified encoding can be computed in time $O_{p,\varepsilon}(n)$ and is ℓ-list decodable in time $\widetilde{O}_{p,\varepsilon}(N)$ within radius $1 - (1-c)\alpha \le \left(1 - \frac{1}{p}\right)(1-\beta)$ from Lemma 4.29. Moreover, from Lemma 4.14, this encoding has an (m, δ)-full-rank partition for $m = \lfloor \frac{\lambda n}{2k} \rfloor$.

Finally, from Theorem 5.4, there exists a randomized oracle data-structure algorithm M that computes Av for any given A, v with probability $2/3$ in time $\widetilde{O}_{p,\varepsilon}(n)$. The number of oracle calls made by M is at most $(N/m) \cdot O\left(\frac{\log n}{c\alpha}\right) \le 2^{\mathsf{poly}(p,1/\varepsilon)} \cdot O(\log n)$. □

A Yao's XOR Lemma for Multi-Output Functions

In this appendix, we present a simple proof of Yao's XOR lemma for multi-output functions.

Theorem A.1 (A restatement of Theorem 2.1)**.** *Let $\mathcal{L}$ be a distribution over $[m]$. Let $f\colon \{0,1\}^n \to \{0,1\}^m$ be a function such that for every circuit C of size s,*

$$\Pr_{\substack{x \sim \{0,1\}^n \\ \ell \sim \mathcal{L}}}[C(x)_\ell = f(x)_\ell] < 1 - \delta.$$

Then for every circuit C' of size s', there exists an index $\ell \in [m]$ such that

$$\Pr_{\bar{x} \sim (\{0,1\}^n)^k}\left[C'(\bar{x})_\ell = f^{\oplus k}(\bar{x})_\ell\right] < \frac{1}{2} + \varepsilon,$$

where $k = O\big(\log(1/\varepsilon)/(\varepsilon\delta)^2\big)$ and $s = O\big((s'+m) \cdot \big(\log(1/\delta)/\varepsilon^2\big)\big)$.

Proof of Theorem A.1. We prove the contrapositive by presenting a non-uniform reduction R from the task of computing f on a $(1-\delta)$-fraction of inputs to the task of computing $f^{\oplus k}$ on $\big(\frac{1}{2} + \varepsilon\big)$-fraction of inputs. Let $\mathcal{O}$ be an oracle such that for all indices $\ell \in [m]$,

$$\Pr_{y \sim (\{0,1\}^n)^k}\left[\mathcal{O}(y)_\ell = f^{\oplus k}(y)_\ell\right] \ge \frac{1}{2} + \varepsilon. \tag{16}$$

Fix an input $x \in \{0,1\}^n$, and consider the following distribution. Let $i \sim [k]$ and $(y_1, \cdots, y_k) \sim (\{0,1\}^n)^k$, and define $r := (i, y_1, \cdots, y_k)$ and $\Gamma(x; r) := (y_1, \cdots, y_{i-1}, x, y_{i+1}, \cdots, y_k)$. Let $\Omega := [k] \times (\{0,1\}^n)^k$ denote the space of r. We also define $\alpha^f(r) := \bigoplus_{j \in [k] \setminus \{i\}} f(y_j)$. Let $\Gamma(x; \text{-})$ denote the distribution of $\Gamma(x; r)$ for $r \sim \Omega$.

We define an oracle algorithm $R^{\mathcal{O}}(x; \bar{r}, \alpha)$ as follows. It takes as input $x \in \{0,1\}^n$, coin flip sequences $\bar{r} = (r_1, \cdots, r_T) \in \Omega^T$, and an advice string $\alpha = (\alpha_1, \ldots, \alpha_T) \in (\{0,1\}^m)^T$. It outputs the bitwise majority of $\mathcal{O}(y_1) \oplus \alpha_1, \cdots, \mathcal{O}(y_T) \oplus \alpha_T$, where $y_1 := \Gamma(x; r_1), \cdots, y_T := \Gamma(x; r_T)$.

We claim that there exists a "Trevisan–Vadhan advice [TV07]" function $\alpha\colon \Omega^T \to \{0,1\}$ such that for all $\ell \in [m]$

$$\Pr_{\substack{x \sim \{0,1\}^n, \\ \bar{r} \sim \Omega^T}}\left[R^{\mathcal{O}}(x; \bar{r}, \alpha(\bar{r}))_\ell = f(x)_\ell\right] \ge 1 - \delta, \tag{17}$$

where T is a parameter chosen later. Fix arbitrary $\ell \in [m]$. Define $\alpha(\bar{r}) \coloneqq (\alpha^f(r_1), \cdots, \alpha^f(r_T))$. By Lemma 3.6 and (16), with probability at least $1 - \delta/2$ over a choice of $x \sim \{0,1\}^n$, it holds that

$$\Pr_{y \sim \Gamma(x;\text{-})}\Big[\mathcal{O}(y)_\ell = f^{\oplus k}(y)_\ell\Big] \geq \frac{1}{2} + \frac{\varepsilon}{2}. \tag{18}$$

We call such an input $x \in \{0,1\}^n$ *good*. We claim that $R^{\mathcal{O}}(x; \bar{r}, \alpha(\bar{r}))_\ell = f(x)_\ell$ for every good $x \in \{0,1\}^n$. By Hoeffding's inequality, with probability at least $1 - \delta/2$ over a choice of $y_1, \cdots, y_T \sim \Gamma(x;\text{-})$, we have

$$\frac{1}{T} \cdot \left|\left\{i \in [T] \;\middle|\; \mathcal{O}(y_i)_\ell = f^{\oplus k}(y_i)_\ell\right\}\right| \geq \Pr_{y \sim \Gamma(x;\text{-})}\Big[\mathcal{O}(y)_\ell = f^{\oplus k}(y)_\ell\Big] - \frac{\varepsilon}{4} \geq \frac{1}{2} + \frac{\varepsilon}{4} \tag{19}$$

for a sufficiently large $T = O(\log(1/\delta)/\varepsilon^2)$. If $\mathcal{O}(y_i)_\ell = f^{\oplus k}(y_i)_\ell$, then $\mathcal{O}(y_i)_\ell \oplus (\alpha_i)_\ell = f^{\oplus k}(y_i) \oplus \alpha^f(r_i)_\ell = f(x)$. Thus, under the event of (19), the majority of $\mathcal{O}(y_i)_\ell \oplus (\alpha_i)_\ell$ over all $i \in [T]$ is equal to $f(x)$. It follows from (18), (19) and a union bound that $R^{\mathcal{O}}(x; \bar{r}, \alpha(\bar{r})) = f(x)$ with probability at least $1 - \delta$ over a choice of x and $\bar{r}$. This completes the proof of (17).

By (17), we have

$$\Pr_{\substack{x \sim \{0,1\}^n \\ \bar{r} \sim \Omega^T \\ \ell \sim \mathcal{L}}}\Big[R^{\mathcal{O}}(x; \bar{r}, \alpha(\bar{r}))_\ell = f(x)_\ell\Big] \geq 1 - \delta.$$

In particular, there exists $\bar{r} \in \Omega^T$ such that

$$\Pr_{\substack{x \sim \{0,1\}^n \\ \ell \sim \mathcal{L}}}\Big[R^{\mathcal{O}}(x; \bar{r}, \alpha(\bar{r}))_\ell = f(x)_\ell\Big] \geq 1 - \delta.$$

It follows that there exists an advice string $(\bar{r}, \alpha(\bar{r}))$ that enables $R^{\mathcal{O}}$ to compute f on a $(1-\delta)$-fraction of inputs. □

B Optimal Reduction for Small Field

Theorem B.1. *Let $\mathbb{F}_p$ be a finite field of prime order p and $\varepsilon > 0$ be a constant. There exists a randomized $O_{p,\varepsilon}(\log n)$-query oracle algorithm $M^{\mathcal{O}}$ that runs in time $\widetilde{O}_{p,\varepsilon}(n^2)$ such that, for any oracle $\mathcal{O}$ satisfying*

$$\mathop{\mathbb{E}}_{A,B \sim \mathbb{F}_p^{n \times n}}[\mathrm{dist}(\mathcal{O}(A,B),\, A \cdot B)] \leq 1 - \frac{1}{p} - \varepsilon,$$

it holds for every sufficiently large n and all $A, B \in \mathbb{F}_p^{n \times n}$ that

$$\Pr_{M^{\mathcal{O}}}\big[M^{\mathcal{O}}(A,B) = A \cdot B\big] \geq 1 - o(1).$$

Notation. For a vector $x \in \mathbb{F}_p^n$ and $\rho \in [0,1]$, let $\mathrm{Ball}(x,\rho) \subseteq \mathbb{F}_p^n$ be the set of vectors $y \in \mathbb{F}_p^n$ that satisfies $\mathrm{dist}(x,y) \leq \rho$. For a matrix $A \in \mathbb{F}_p^{n \times n}$, let $\mathsf{vec}(A) \in \mathbb{F}_p^{n^2}$ be the vector representation of A, that is, for any $i, j \in [n]$, the $(n \cdot (i-1) + j)$-th entry of $\mathsf{vec}(A)$ is given by $A_{i,j}$. For a graph $G = ([n], E)$ and $k \geq 0$, let $W \subseteq [n]^k$ be the k-walk collection on G. For a walk $w =$

$(u_1, \dots, u_k) \in W$, let $\mathsf{visit}(w) = \{u_1, \dots, u_k\}$. A subset $P \subseteq W$ of walks is said to be *vertex-disjoint* if $\mathsf{visit}(w) \cap \mathsf{visit}(w') = \emptyset$ for any distinct pair of walks $w, w' \in W$.

For two graphs $G_i = ([n_i], E_i)$ $(i = 1, 2)$, the tensor product $G_1 \otimes G_2$ is the graph on vertex set $[n_1] \times [n_2]$ such that two vertices $(u_1, u_2), (v_1, v_2) \in [n_1] \times [n_2]$ are adjacent on $G_1 \otimes G_2$ if and only if $\{u_i, v_i\} \in E_i$ for both $i = 1, 2$. By definition, if $(i_1, \dots, i_k)$ and $(j_1, \dots, j_k)$ form k-walks on G, then $((i_1, j_1), \dots, (i_k, j_k))$ forms a k-walk on $G \otimes G$. Conversely, if $((i_1, j_1), \dots, (i_k, j_k))$ forms a k-walk on $G \otimes G$, then both $(i_1, \dots, i_k)$ and $(j_1, \dots, j_k)$ form k-walks on G. It is well known that, if G is d-regular and λ-expander, then $G \otimes G$ is d^2-regular and λ-expander.[11]

B.1 Approximate List-Decodable Codes

We consider encodings that are *approximate list-decoding*, which is a relaxed notion of the list-decoding of Definition 4.3. Recall that, in the list-decoding within radius ρ, given a noisy codeword $\tilde{y} \in \mathbb{F}_p^N$, we are asked to output a set of vectors $L = \{x_1, \dots, x_\ell\} \in \mathbb{F}_p^n$ such that for any codeword $y = \mathsf{Enc}(x) \in \mathrm{Ball}(\tilde{y}, \rho)$, it holds that $x \in L$. In the δ-approximate list-decoding within radius ρ, given a noisy codeword $\tilde{y} \in \mathbb{F}_p^N$, our task is to find a set of vectors $L = \{x_1, \dots, x_\ell\} \in \mathbb{F}_p^n$ such that, for any codeword $y = \mathsf{Enc}(x) \in \mathrm{Ball}(\tilde{y}, \rho)$, it holds that $x \in \mathrm{Ball}(x_i, \delta)$ for some $x_i \in L$. Thus, the approximate list-decoding for $\delta = 0$ corresponds to the exact version. The notion of approximate list-decodabliity was previously considered in [DHKNT21; IJKW10] in the literature of direct product encoding.

Definition B.2 (approximate list-decoding)**.** *An encoding* $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ *is* δ-approximate ℓ-list-decodable within radius ρ *if, for any* $\tilde{y} \in \mathbb{F}_p^N$*, there exists a set* $L = \{x_1, \dots, x_\ell\} \subseteq \mathbb{F}_p^n$ *such that, for any* $y = \mathsf{Enc}(x) \in \mathrm{Ball}(\tilde{y}, \rho)$*, we have* $\mathrm{Ball}(x, \delta) \cap L \neq \emptyset$*. An algorithm that computes such set* L *given* $\tilde{y}$ *as input is called an* approximate list-decoding algorithm.

Jeronimo, Srivastava, and Tulsiani [JST21] and Jeronimo [Jer23] implicitly presented a randomized approximate list-decoding algorithm for direct sum encoding with respect to a collection $W \subseteq [n]^k$ of k tuples that is *splittable*, which is a certain kind of expansion property of the set of k-tuples. Since expander walks satisfy the splittable property, we immediately obtain the following.

Lemma B.3 (Implicit in the proof of [Jer23, Theorem 7.5])**.** *Let* $p \geq 2$ *be a constant prime,* $\delta, \beta, \lambda > 0$ *and* $k \in \mathbb{N}$ *be any constants such that*

$$\beta \geq \max\left\{2^{10}\sqrt{\lambda k^3},\ 4\left(1 - \frac{(1 - \cos(\pi/p))^2\delta^2}{4}\right)^{k/2}\right\}. \tag{20}$$

Let $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^W$ *be the direct sum encoding with respect to the* k*-walk collection* $W \subseteq [n]^k$ *on a regular* λ*-expander graph. Then, for some* $\ell = O_{p,k,\beta}(1)$*, the encoding* Enc *is* δ*-approximate* ℓ*-list-decodable within radius* $(1 - 1/p)(1 - \beta)$ *by an* $\widetilde{O}_{p,k,\beta}(|W|)$*-time randomized algorithm.*

B.2 Exact-to-Approximate Reduction

We show that the k-walk collection on a regular graph can be partitioned into $O(1)$ subsets such that each subset in the partition is vertex-disjoint.

[11] More generally, if A has eigenvalues $\lambda_1, \dots, \lambda_n$ and B has eigenvalues $\mu_1, \dots, \mu_n$, then the eigenvalues of the tensor product $A \otimes B$ is given by $\lambda_i \mu_j$ for $i, j \in [n]$ [HJ91].

Lemma B.4. *Let $G = ([n], E)$ be a d-regular graph and $W \subseteq [n]^k$ be the k-walk collection on G. Let $a = k^2 d^{k-1} + 1$. Then, there exists a partition $\mathcal{P} = P_1 \sqcup \cdots \sqcup P_a$ of W such that, for every $i \in [a]$, the set P_i is vertex-disjoint. Moreover, we can compute such partition in time $O(|W|)$.*

Proof. For the k-walk collection W, define the intersection graph I with respect to W by

- The vertex set is given by $V(I) = W$, and
- Any pair of distinct walks $w, w' \in W$ forms an edge of I if and only if $\mathsf{visit}(w) \cap \mathsf{visit}(w') \neq \emptyset$.

To distinguish vertices of G and vertices of I, we call the former a *G-vertex* and the latter an *I-vertex*. Fix an I-vertex $w \in V(I)$. Since $|\mathsf{visit}(w)| \leq k$ and every G-vertex $v \in V(G)$ is visited by at most $\sum_{i=0}^{k-1} d^i \cdot d^{k-1-1} = k \cdot d^{k-1}$ walks in W, the degree of the I-vertex w on I is at most $k \cdot k d^{k-1}$. Therefore, the maximum degree of I is at most $k^2 d^{k-1}$ and there exists a proper vertex-coloring $\chi \colon V(I) \to [a]$, where $a = k^2 d^{k-1} + 1$. This can be computed by $O(|W|)$ time by the straightforward greedy algorithm (order the vertices W and then color them one by one using the smallest possible color).

For each $i \in [a]$, let $P_i = \chi^{-1}(i) \subseteq W$. Since χ is a proper coloring, each P_i forms an independent set in I (that is, there are no edges of I between two vertices in P_i). In other words, each P_i is vertex-disjoint. □

Corollary B.5. *Let $G = ([n], E)$ be a d-regular graph with girth at least k and $W \subseteq [n]^k$ be the k-walk collection on G. Let $a = k^2 d^{k-1} + 1$. Then, there exists a partition $\mathcal{P} = P_1 \sqcup \cdots \sqcup P_a \sqcup P'$ of W such that*

- *For every $i \in [a]$, the set P_i is vertex-disjoint and every $w \in P_i$ satisfies $\mathsf{visit}(w) = k$.*
- *The set P' in the partition satisfies $|P'| \leq \frac{k}{d} \cdot |W|$.*

Moreover, we can compute such partition in time $O(|W|)$.

Proof. Let $\mathcal{P}' = P'_1 \sqcup \cdots \sqcup P'_a$ be the partition of Lemma B.4. For every $i \in [a]$ let $P_i \subseteq P'_i$ be the set of walks $w \in P'_i$ such that $|\mathsf{visit}(w)| = k$. Finally, let $P' = W \setminus (P_1 \cup \cdots \cup P_a)$. Clearly, each P_i satisfies the condition of Corollary B.5. We bound $|P'|$. Note that the set P' consists of walks $w \in W$ such that $|\mathsf{visit}(w)| < k$. We say that a walk $w = (u_1, \ldots, u_k)$ is *backtracking* if $u_{j-1} = u_{j+1}$ for some $j = 2, \ldots, k-1$. By the union bound over $j = 2, \ldots, k-1$, a random walk $w \sim W$ is backtracking with probability at most k/d. Since G has girth at least k, if w is not backtracking, then $|\mathsf{visit}(w)| = k$. Therefore, we have

$$\begin{aligned}
\frac{|P'|}{|W|} &= \Pr_{w \sim W}[|\mathsf{visit}(w)| < k] \\
&\leq \Pr_{w \sim W}[w \text{ is backtracking}] \\
&\leq \frac{k}{d}.
\end{aligned}$$

□

Next, we define the lifting of a matrix.

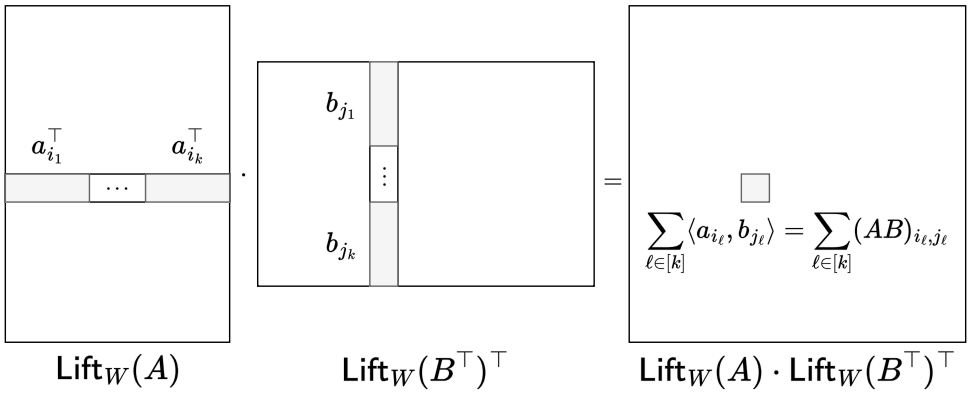

Figure 4: The $(\mathbf{i},\mathbf{j})$-th entry of $\mathsf{Lift}_W(A)\cdot\mathsf{Lift}_W(B)$ for $\mathbf{i}=(i_1,\ldots,i_k)$ and $\mathbf{j}=(j_1,\ldots,j_k)$ is equal to $\sum_{\ell=1}^{k}(A\cdot B)_{i_\ell,j_\ell}$.

Definition B.6 (W-lifting of matrices)**.** *Let $W\subseteq[n]^k$ be a collection of k-tuples. For a matrix $A\in\mathbb{F}_p^{n\times m}$ with row vectors $a_1,\ldots,a_n\in\mathbb{F}_p^m$, the W-*lifting *of A is the matrix $\mathsf{Lift}_W(A)\in\mathbb{F}_p^{W\times km}$ whose $\mathbf{i}=(i_1,\ldots,i_k)$-th row vector (for $\mathbf{i}\in W$) is the concatenation of $a_{i_1},\ldots,a_{i_k}$. That is, for each $\mathbf{i}=(i_1,\ldots,i_k)\in W$, $s\in[k]$ and $t\in[m]$, the $(\mathbf{i},(s-1)m+t)$-th entry of $\mathsf{Lift}_W(M)$ is given by $(a_{i_s})_t$.*

Note that the matrix A is not necessarily be a square matrix in Definition B.6. See Fig. 4 to gain some intuition of W-lifting. In the following, we show a useful relation between W-lifting and direct sum encoding (Definition 4.9).

Lemma B.7. *Let $G=([n],E)$ be a d-regular graph and $W\subseteq[n]^k$ be the collection of k-tuples. For a matrix $A\in\mathbb{F}_p^{n\times m}$, let $\mathsf{Lift}_W(A)\in\mathbb{F}_p^{W\times km}$ be the lifting of A with respect to W. Then, for any $A\in\mathbb{F}_p^{n\times m}$ and $B\in\mathbb{F}_p^{m\times n}$, we have*

$$\mathsf{vec}\Big(\mathsf{Lift}_W(A)\cdot\mathsf{Lift}_W(B^\top)^\top\Big)=\mathsf{Enc}(\mathsf{vec}(A\cdot B)),$$

where $\mathsf{Enc}\colon\mathbb{F}_p^{n^2}\to\mathbb{F}_p^{|W|^2}$ is the direct sum encoding with respect to the k-walk collection on the tensor product $G\otimes G$.

Proof. For two matrices $A,B\in\mathbb{F}_p^{n\times n}$, consider $\mathsf{Lift}_W(A)\cdot\mathsf{Lift}_W(B)^\top\in\mathbb{F}_p^{W\times W}$. Let $a_1^\top,\ldots,a_n^\top\in\mathbb{F}_p^n$ be the row vectors of A and $b_1,\ldots,b_n\in\mathbb{F}_p^n$ be the column vectors of B. By definition of W-lifting, the $(\mathbf{i},\mathbf{j})$-element of $\mathsf{Lift}_W(A)\cdot\mathsf{Lift}_W(B)^\top$ for $\mathbf{i}=(i_1,\ldots,i_k)$ and $\mathbf{j}=(j_1,\ldots,j_k)$ can be written as

$$\begin{aligned}\Big(\mathsf{Lift}_W(A)\cdot\mathsf{Lift}_W(B^\top)^\top\Big)_{\mathbf{i},\mathbf{j}}&=\sum_{s\in[k],t\in[n]}(a_{i_s})_t\cdot(b_{j_s})_t\\&=\sum_{s\in[k]}\langle a_{i_s},b_{j_s}\rangle\\&=\sum_{s\in[k]}(A\cdot B)_{i_s,j_s}.\end{aligned}$$

In other words, the $(\mathbf{i},\mathbf{j})$-th entry of $\mathsf{Lift}_W(A)\cdot\mathsf{Lift}_W(B^\top)^\top$ is equal to the sum of the $(i_1,j_1),\dots,(i_k,j_k)$-th entries of $A\cdot B$. Since $\mathbf{i}$ and $\mathbf{j}$ form a k-walk on G, the sequence of pairs $(i_1,j_1),\dots,(i_k,j_k)$ forms a k-walk on the tensor product $G\otimes G$. Therefore, the vectorization of $\mathsf{Lift}_W(A)\cdot\mathsf{Lift}_W(B^\top)^\top$ is the direct sum encoding of $A\cdot B$ with respect to the k-walk collection W' on $G\otimes G$. □

In the following, we show that an average-case approximation solver can be used to compute an approximate codeword of the direct sum encoding with respect to the k-walk collection on some expander graph.

Lemma B.8. *Let $\mathbb{F}_p$ be the finite field of order p and $\varepsilon>0$. There exists a randomized oracle algorithm M such that, for every oracle $\mathcal{O}$ such that for every sufficiently large n,*

$$\mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ \mathcal{O}}}[\mathrm{dist}(\mathcal{O}(A,B),\,A\cdot B)]\le 1-\alpha,$$

given two matrices $A,B\sim\mathbb{F}_p^{n\times n}$, a positive integer $k>0$, and a d-regular graph $G=([n],E)$ whose girth is at least k, it holds for sufficiently large n that

$$\mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M^{\mathcal{O}}}}\left[\mathrm{dist}\left(M^{\mathcal{O}}(A,B,G,k),\,\mathsf{Enc}(\mathsf{vec}(A\cdot B))\right)\right]\le 1-\alpha+\frac{k^2}{d^2},\tag{21}$$

where $\mathsf{Enc}\colon\mathbb{F}_p^{n^2}\to\mathbb{F}_p^{(nd^{k-1})^2}$ is the direct sum encoding with respect to the k-walk collection on $G\otimes G$. The algorithm M runs in time $O_{k,d}(n^2)$ and makes $O_{k,d}(1)$ queries.

Proof. The oracle algorithm $M^{\mathcal{O}}$ run in input $A,B\in\mathbb{F}_p^{n\times n}$, $k>0$, and $G=([n],E)$ as follows:

1. Let $W\subseteq[n]^k$ be the k-walk collection on G and compute $\mathsf{Lift}_W(A),\mathsf{Lift}_W(B^\top)\in\mathbb{F}_p^{W\times kn}$.
2. Compute the partition $\mathcal{P}=P_1\sqcup\cdots\sqcup P_a\sqcup P'$ of Corollary B.5. Suppose that $|P_i|\le kn$ for all $i\in[a]$ (otherwise, partition P_i into $O(1)$ subsets each of size at most kn).
3. For every $i,j\in[a]$, run the algorithm of Lemma 4.24 on input $\mathsf{Lift}_W(A)|_{P_i,\star},\mathsf{Lift}_W(B)|_{\star,P_j}$ using $\mathcal{O}$ as oracle. Let $\widetilde{C}_{i,j}\in\mathbb{F}_p^{|P_i|\times|P_j|}$ be the input.
4. Output a matrix $\widetilde{C}\in\mathbb{F}_p^{W\times W}$ that can be obtained by aligning $\widetilde{C}_{i,j}$ for all $i,j\in[a]$ and filling the rest entries (i.e., entries whose either row or column index is in P') arbitrary.

Note that M runs in time $O_{k,d}(n^2)$ and makes at most $a^2=O_{k,d}(1)$ queries. We prove Eq. (21). Let $\mathcal{P}=P_1\sqcup\cdots\sqcup P_a\sqcup P'$ be the partition computed at Step 2.

For each $i\in[a]$, the set P_i consists of vertex-disjoint walks and $\mathsf{visit}(w)=k$; thus the row vectors of $\mathsf{Lift}_W(A)$ corresponding to walks in P_i are independent and uniformly distributed over $\mathbb{F}_p^{kn}$ when $A\sim\mathbb{F}_p^{n\times n}$ is a uniformly random matrix. Therefore, for each $i,j\in[a]$, when $A,B\sim\mathbb{F}_p^{n\times n}$, the marginal distribution of $\left(\mathsf{Lift}_W(A)|_{P_i,\star},\mathsf{Lift}_W(B)|_{\star,P_j}\right)$ is uniform over $\mathbb{F}_p^{|P_i|\times kn}\times\mathbb{F}_p^{kn\times|P_j|}$. Since $|P_i|,|P_j|\le kn$, from Lemma 4.24, for each $i,j\in[a]$, we have

$$\mathbb{E}\left[\mathrm{dist}\left(\widetilde{C}_{i,j},\mathsf{Lift}_W(A)|_{P_i}\cdot\mathsf{Lift}_W(B)|_{P_j}\right)\right]\le 1-\alpha,$$

where $\widetilde{C}_{i,j}$ is the matrix computed at Step 3 and the expectation is taken over $A, B \sim \mathbb{F}_p^{n\times n}$ and the internal randomness of the oracle algorithm of Lemma 4.24.

Therefore, we have

$$\begin{aligned}
\text{LHS of Eq. (21)} &= \mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M^{\mathcal{O}}}}\Big[\mathrm{dist}\Big(\widetilde{C},\, \mathsf{Lift}_W(A)\cdot \mathsf{Lift}_W(B^\top)^\top\Big)\Big] && \because \text{Lemma B.7}\\
&\le \mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M^{\mathcal{O}}\\ i,j\sim[a]}}\Big[\mathrm{dist}\Big(\widetilde{C}_{i,j},\, \mathsf{Lift}_W(A)|_{P_i}\cdot \mathsf{Lift}_W(B)|_{P_j}\Big)\Big] + \frac{|P'|^2}{|W|^2}\\
&\le 1-\alpha+\frac{k^2}{d^2}.
\end{aligned}$$

□

We prove that the proximity of $A\cdot B$ and C can be efficiently checked given $A, B, C \in \mathbb{F}_p^{n\times n}$ as input.

Lemma B.9. *For any $\eta > 0$, there exists an $O(n^2 \log n/\eta)$-time randomized algorithm M that, on input $A, B, C \in \mathbb{F}_p^{n\times n}$, satisfies the following:*

- *If $\mathrm{dist}(A\cdot B, C) \le \eta$, then M outputs Yes with probability $1 - 1/n^3$.*
- *If $\mathrm{dist}(A\cdot B, C) \ge 2\eta$, then M outputs No with probability $1 - 1/n^3$.*

Proof. The algorithm M runs on input $A, B, C \in \mathbb{F}_p^{n\times n}$, repeat checking if $(A\cdot B)_{i,j} = C_{i,j}$ for uniformly random $i, j \sim [n]$ for $T = O\Big(\frac{\log n}{\eta}\Big)$ times. If the number of iterations at which $(A\cdot B)_{i,j} = C_{i,j}$ holds is at least $(1 - 1.5\eta)\cdot T$, output Yes. Otherwise, output No.

If $\mathrm{dist}(A\cdot B, C) \le \eta$, then $(A\cdot B)_{i,j} = C_{i,j}$ with probability at least $1-\eta$ over the choice of $i, j \sim [n]$. Therefore, by the Chernoff bound, the algorithm M outputs Yes with probability $1 - \exp(-\eta T/12) \ge 1 - 1/n^3$. The case of $\mathrm{dist}(A\cdot B, C) \ge 2\eta$ is the same. □

Lemma B.10. *Let $\mathbb{F}_p$ be a finite field of prime order p and $\delta, \varepsilon > 0$ be constants. There exists a randomized $O(\log n)$-query oracle algorithm $M^{\mathcal{O}}$ that runs in time $\widetilde{O}(n^2)$ such that, for any oracle $\mathcal{O}$ satisfying*

$$\mathop{\mathbb{E}}_{A,B\sim\mathbb{F}_p^{n\times n}}[\mathrm{dist}(\mathcal{O}(A,B),\, A\cdot B)] \le 1 - \frac{1}{p} - \varepsilon,$$

it holds for every sufficiently large n that

$$\Pr_{A,B\sim\mathbb{F}_p^{n\times n}}\Big[\Pr_{M^{\mathcal{O}}}\big[\mathrm{dist}\big(M^{\mathcal{O}}(A,B), A\cdot B\big) \le \delta\big] \ge 1 - o(1)\Big] \ge \frac{\varepsilon}{4}.$$

Proof. Fix a randomized oracle $\mathcal{O}$. For given $p \ge 2$, $\delta, \varepsilon > 0$, we set constants $k \in \mathbb{N}$ and $\lambda > 0$ such that Eq. (20) holds for $\beta = \varepsilon/4$. Note that we can set $k = O(p^4 \log(1/\varepsilon)/\delta^2)$ and $\lambda = O(\varepsilon^2/k^3)$. For $d = \Big\lceil \max\Big\{\frac{4}{\lambda^2}, \frac{2k}{\sqrt{\varepsilon}}\Big\}\Big\rceil$, fix a d-regular λ-expander graph $G = ([n], E)$ with girth at least k. From Lemma 4.12, with probability $\Omega(1)$ a random regular graph $G_{n,d}$ satisfies this property. Thus, we

can construct such G in time $O(n)$ with probability $1-o(1)$. From Lemma B.8 for $\alpha = \frac{1}{p}+\varepsilon$, there exists a randomized oracle algorithm $M_0^{\mathcal{O}}$ such that

$$\begin{aligned}\mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M_0^{\mathcal{O}}}}\left[\operatorname{dist}\left(M_0^{\mathcal{O}}(G,k,A,B),\,\mathsf{Enc}(\mathsf{vec}(A\cdot B))\right)\right] &\le 1-\frac{1}{p}-\varepsilon+\frac{k^2}{d^2}\\ &\le 1-\frac{1}{p}-\frac{3\varepsilon}{4}. \qquad \because\ d^2\ge 4k/\varepsilon\end{aligned}$$

Call an instance $(A,B)\in\mathbb{F}_p^{n\times n}\times\mathbb{F}_p^{n\times n}$ *good* if

$$\mathop{\mathbb{E}}_{M_0^{\mathcal{O}}}\left[\operatorname{dist}\left(M_0^{\mathcal{O}}(G,k,A,B),\,\mathsf{Enc}(\mathsf{vec}(A\cdot B))\right)\right]\le 1-\frac{1}{p}-\frac{\varepsilon}{2}.$$

By Markov's inequality (Lemma 4.1), at least $\varepsilon/4$-fraction of instances (A,B) are good.

We present an algorithm that outputs a matrix that is $(1-\delta)$-close to $A\cdot B$ for every good (A,B). Let $\widetilde{C}\in\mathbb{F}_p^{W\times W}$ be the output of $M_0^{\mathcal{O}}$ on input (A,B), where $W\subseteq[n]^k$ is the k-walk collection on G. By Markov's inequality, with probability $\varepsilon/4$ over the internal randomness of $M_0^{\mathcal{O}}$, we have $\operatorname{dist}(\widetilde{C},\mathsf{Enc}(\mathsf{vec}(A\cdot B))))\le 1-\frac{1}{p}-\frac{\varepsilon}{4}$. Note that the encoding Enc is the direct sum encoding with respect to the k-walk collection on $G\otimes G$, which is a d^2-regular λ-expander graph. By our choice of parameters k and λ, from Lemma B.3, this encoding Enc is δ-approximate ℓ-list decodable within radius $\left(1-\frac{1}{p}\right)(1-\beta)\le 1-\frac{1}{p}-\frac{\varepsilon}{4}$ by an $\widetilde{O}(n^2)$-time algorithm. Thus, by running the list-decoding algorithm on input $\mathsf{vec}\left(\widetilde{C}\right)\in\mathbb{F}_p^{|W|^2}$, we obtain a set of matrices $C_1,\dots,C_\ell$ for some $\ell=O(1)$ that contains a matrix that is δ-close to $A\cdot B$ in time $\widetilde{O}(n^2)$. From Lemma B.3, we can identify such matrices from the set in time $\widetilde{O}(n^2)$. □

Lemma B.11. *Let $\mathbb{F}_p$ be a finite field of prime order p and $\gamma,\varepsilon>0$ be constants. There exists a randomized $O(\log n)$-query $\widetilde{O}(n^2)$-time oracle algorithm $M^{\mathcal{O}}$ such that, for any oracle $\mathcal{O}$ satisfying*

$$\mathop{\mathbb{E}}_{A,B\sim\mathbb{F}_p^{n\times n}}\left[\operatorname{dist}(\mathcal{O}(A,B),\,A\cdot B)\right]\le 1-\frac{1}{p}-\varepsilon,$$

it holds for every sufficiently large n that

$$\mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M^{\mathcal{O}}}}\left[\operatorname{dist}\left(M^{\mathcal{O}}(A,B),\,A\cdot B\right)\right]\le\gamma.$$

Proof. Let $A,B\in\mathbb{F}_p^{n\times n}$ be the input to $M^{\mathcal{O}}$. Let $t=c\log(1/\varepsilon)/(\gamma^2\varepsilon^2)$ and $m=\lfloor n/t\rfloor$ for a sufficiently large constant c. Let $I_1\sqcup\cdots\sqcup I_t\sqcup I'$ be the partition of $[n]$ defined by

$$I_1=\{1,\dots,m\},\quad I_2=\{m+1,\dots,2m\},\quad \dots,\quad I_t=\{(t-1)m,\dots,tm\},\quad I'=\{tm+1,\dots,n\}.$$

For $i,j\in[t]$, let $A_i=A|_{I_i,[n]}$ and $B_j=B|_{[n],I_j}$.

From Lemma B.10 for $\delta=\frac{\gamma}{4t^2}$, there exists an oracle algorithm $M_0^{\mathcal{O}}$ such that

$$\Pr_{A,B\sim\mathbb{F}_p^{n\times n}}\left[\Pr_{M_0^{\mathcal{O}}}\left[\operatorname{dist}\left(M_0^{\mathcal{O}}(A,B),\,A\cdot B\right)\le\delta\right]\ge 1-o(1)\right]\ge\frac{\varepsilon}{4}.$$

Our reduction $M^{\mathcal{O}}$ runs on input A,B as follows:

1. Let $D \in \mathbb{F}_p^{n\times n}$ be a matrix that is initialized to be the all-zero matrix.
2. For each $i, j \in [t]$, repeat the following for $O(\log(1/\gamma)/\varepsilon)$ times:
   (a) Sample $\overline{A}, \overline{B} \sim \mathbb{F}_p^{n\times n}$.
   (b) Replace $\overline{A}|_{I_i,[n]}$ by A_i. Similarly, replace $\overline{B}|_{[n],I_j}$ by B_j.
   (c) Run the oracle algorithm $M_0^{\mathcal{O}}(\overline{A}, \overline{B})$. Let $C \in \mathbb{F}_p^{n\times n}$ be the output.
   (d) If $C|_{I_i,I_j}$ is δ-close to $A_i \cdot B_j$ (which can be checked by Lemma B.9), replace $D|_{I_i,I_j}$ by $C|_{I_i,I_j}$.
3. Output D.

We prove the correctness of $M_1^{\mathcal{O}}$. We have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M^{\mathcal{O}}}}\left[\operatorname{dist}\left(M^{\mathcal{O}}(A,B),\, A\cdot B\right)\right] &\le \mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M^{\mathcal{O}}\\ i,j\sim[t]}}\left[\operatorname{dist}\left(M^{\mathcal{O}}(A,B)|_{I_i,I_j},\, (A\cdot B)|_{I_i,I_j}\right)\right] + 2\frac{|I'|^2}{n^2} \\
&\le \mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M^{\mathcal{O}}}}\left[\operatorname{dist}\left(M^{\mathcal{O}}(A,B)|_{I_1,I_1},\, A_1\cdot B_1\right)\right] + \frac{2}{t^2} \\
&\le \mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M^{\mathcal{O}}}}\left[\operatorname{dist}\left(M^{\mathcal{O}}(A,B)|_{I_1,I_1},\, A_1\cdot B_1\right)\right] + \frac{\gamma}{4}. \qquad (22)
\end{aligned}$$

To bound the right most term, let $C \in \mathbb{F}_p^{n\times n}$ be the matrix obtained at Step 3(c) at an iteration where $i = j = 1$. Call an instance $(\overline{A}, \overline{B}) \in \left(\mathbb{F}_p^{n\times n}\right)^2$ *good* if $\mathbb{E}_{M_0^{\mathcal{O}}}[\operatorname{dist}\left(M_0^{\mathcal{O}}(A,B),\, A\cdot B\right)] \le \delta$. Note that at least $\varepsilon/4$-fraction of $(\overline{A}, \overline{B})$ are good. From Lemma 3.6 (for $x = (A_1, B_1)$, $y = (\overline{A}, \overline{B})$, and S being the indicator of the set of good instances), for at least $(1-\gamma/4)$-fraction of $(A_1, B_1) \sim \mathbb{F}_p^{m\times n} \times \mathbb{F}_p^{n\times m}$, the probability that $(\overline{A}, \overline{B})$ obtained at Step 2(b) is good with probability at least $\frac{1}{2}\cdot\frac{\varepsilon}{4} = \frac{\varepsilon}{8}$. For such (A_1, B_1), during the $O(\log(1/\gamma)/\varepsilon)$ iterations of Step 2 (at $i = j = 1$), at least one instance $(\overline{A}, \overline{B})$ is good (and thus the matrix C is δ-close to $A\cdot B$) with probability $1-\gamma/4$. If C is δ-close to $A\cdot B$, then the matrix $C|_{I_1,J_1}$ agrees with $A_1\cdot B_1$ on at least $n^2/t^2 - \delta n^2$ entries; thus $\operatorname{dist}(C|_{I_1,J_1},\, A_1\cdot B_1) \le t^2\delta^2 \le \gamma/4$. Therefore, we have

$$\begin{aligned}
\text{Eq. (22)} &= \mathop{\mathbb{E}}_{\substack{A_1,B_1\sim\mathbb{F}_p^{m\times n}\times\mathbb{F}_p^{n\times m}\\ \overline{A},\overline{B}\\ M_0^{\mathcal{O}}}}[\operatorname{dist}(C|_{I_1,J_1}, A_1\cdot B_1)] + \frac{\gamma}{4} \\
&\le \underbrace{\frac{\gamma}{4}}_{\text{sampler property}} + \underbrace{\frac{\gamma}{4}}_{\substack{\text{no } (\overline{A},\overline{B}) \text{ is good}\\ \text{during repetition}}} + \underbrace{\frac{\gamma}{4}}_{\operatorname{dist}\left(C|_{I_1,J_1}, A_1\cdot B_1\right)} + \frac{\gamma}{4} \\
&\le \gamma.
\end{aligned}$$

□

Proof of Theorem B.1. From Lemma B.11 for $\gamma = 1/9$, there exists an oracle algorithm $M_0^{\mathcal{O}}$ such that

$$\mathop{\mathbb{E}}_{\substack{A\sim\mathbb{F}_p^{n\times n}\\ B\sim\mathbb{F}_p^{n\times n}\\ \mathcal{O}}}[\mathrm{dist}(\mathcal{O}(A,B),AB)] \leq \frac{1}{9}.$$

Then, from Corollary 3.3, we obtain the algorithm as desired. □

References

[ADWXXZ25] J. Alman, R. Duan, V. V. Williams, Y. Xu, Z. Xu, and R. Zhou. "More asymmetry yields faster matrix multiplication". en. In: *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2025, pp. 2005–2039. DOI: 10.1137/1.9781611978322.63 (cit. on p. 1).

[AFKLM24] A. Abboud, N. Fischer, Z. Kelley, S. Lovett, and R. Meka. "New Graph Decompositions and Combinatorial Boolean Matrix Multiplication Algorithms". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2024, pp. 935–943. DOI: 10.1145/3618260.3649696 (cit. on p. 1).

[AGGS22] V. R. Asadi, A. Golovnev, T. Gur, and I. Shinkar. "Worst-case to average-case reductions via additive combinatorics". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2022, pp. 1566–1574. DOI: 10.1145/3519935.3520041 (cit. on pp. 1, 3, 4).

[Ale05] M. Alekhnovich. "Linear Diophantine equations over polynomials and soft decoding of Reed–Solomon codes". en. In: *IEEE transactions on information theory* 51 (7 2005), pp. 2257–2265. DOI: 10.1109/tit.2005.850097 (cit. on pp. 9, 31).

[BLR93] M. Blum, M. Luby, and R. Rubinfeld. "Self-Testing/Correcting with Applications to Numerical Problems". In: *J. Comput. Syst. Sci.* 47.3 (1993), pp. 549–595. DOI: 10.1016/0022-0000(93)90044-W (cit. on pp. 1, 11).

[Bor15] C. Bordenave. "A new proof of Friedman's second eigenvalue Theorem and its extension to random lifts". In: *arXiv [math.CO]* (2015). eprint: 1502.04482 (math.CO) (cit. on p. 20).

[BW12] N. Bansal and R. Williams. "Regularity Lemmas and Combinatorial Algorithms". In: *Theory Comput.* 8.1 (2012), pp. 69–94. DOI: 10.4086/TOC.2012.V008A004 (cit. on p. 1).

[CADMGASCMS23] P. J. Coles, M. Aifer, K. Donatella, D. Melanson, M. H. Gordon, T. D. Ahle, D. Simpson, G. Crooks, A. J. Martinez, and F. M. Sbahi. "Thermodynamic AI and Thermodynamic Linear Algebra". In: *Machine Learning with New Compute Paradigms*. 2023 (cit. on p. 1).

[DHKNT21] I. Dinur, P. Harsha, T. Kaufman, I. L. Navon, and A. Ta-Shma. "List-Decoding with Double Samplers". In: *SIAM Journal on Computing* 50 (2 2021), pp. 301–349. DOI: 10.1137/19M1276650 (cit. on p. 41).

[DWZ23] R. Duan, H. Wu, and R. Zhou. "Faster Matrix Multiplication via Asymmetric Hashing". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2023, pp. 2129–2138. DOI: 10.1109/FOCS57990.2023.00130 (cit. on p. 1).

[Fid72] C. M. Fiduccia. "Polynomial evaluation via the division algorithm the fast Fourier transform revisited". In: *Proceedings of the fourth annual ACM symposium on Theory of computing - STOC '72*. the fourth annual ACM symposium (Denver, Colorado, United States). 1972. DOI: 10.1145/800152.804900 (cit. on p. 31).

[Fre79] R. Freivalds. "Fast probabilistic algorithms". In: *Mathematical Foundations of Computer Science (MFCS)*. 1979, pp. 57–69. DOI: 10.1007/3-540-09526-8_5 (cit. on p. 29).

[Fri03] J. Friedman. "A proof of alon's second eigenvalue conjecture". In: *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing* (San Diego, CA, USA). STOC '03. 2003, pp. 720–724. DOI: 10.1145/780542.780646 (cit. on p. 20).

[GG13] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 2013. DOI: 10.1017/cbo9781139856065 (cit. on pp. 9, 31).

[GGR11] P. Gopalan, V. Guruswami, and P. Raghavendra. "List Decoding Tensor Products and Interleaved Codes". In: *SIAM J. Comput.* 40.5 (2011), pp. 1432–1462. DOI: 10.1137/090778274 (cit. on p. 8).

[GNW11] O. Goldreich, N. Nisan, and A. Wigderson. "On Yao's XOR-Lemma". In: *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*. Springer, 2011, pp. 273–301. DOI: 10.1007/978-3-642-22670-0_23 (cit. on pp. 4, 5).

[GSS24] A. Gola, I. Shinkar, and H. Singh. "Matrix Multiplication Reductions". In: *Proceedings of the Approximation, Randomization, and Combinatorial Optimization (APPROX/RANDOM)*. 2024, 34:1–34:15. DOI: 10.4230/LIPICS.APPROX/RANDOM.2024.34 (cit. on pp. 1, 4, 5, 11).

[HJ91] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991. DOI: 10.1017/CBO9780511840371 (cit. on p. 41).

[HKNS15] M. Henzinger, S. Krinninger, D. Nanongkai, and T. Saranurak. "Unifying and Strengthening Hardness for Dynamic Problems via the Online Matrix-Vector Multiplication Conjecture". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2015, pp. 21–30. DOI: 10.1145/2746539.2746609 (cit. on p. 3).

[HLS22] M. Henzinger, A. Lincoln, and B. Saha. "The Complexity of Average-Case Dynamic Subgraph Counting". In: *Proceedings of the Symposium on Discrete Algorithms (SODA)*. 2022, pp. 459–498. DOI: 10.1137/1.9781611977073.23 (cit. on p. 3).

[HLW06] S. Hoory, N. Linial, and A. Wigderson. "Expander graphs and their applications". en. In: *Bulletin of the American Mathematical Society* 43 (4 2006), pp. 439–561. DOI: 10.1090/S0273-0979-06-01126-8 (cit. on p. 21).

[HS22] S. Hirahara and N. Shimizu. "Hardness Self-Amplification from Feasible Hard-Core Sets". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2022, pp. 543–554. DOI: 10.1109/FOCS54457.2022.00058 (cit. on p. 4).

[HS23] S. Hirahara and N. Shimizu. "Hardness Self-Amplification: Simplified, Optimized, and Unified". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2023, pp. 70–83. DOI: 10.1145/3564246.3585189 (cit. on pp. 1, 4, 8, 13, 24, 27, 33, 34, 36).

[HS24] S. Hirahara and N. Shimizu. "Planted Clique Conjectures Are Equivalent". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2024, pp. 358–366. DOI: 10.1145/3618260.3649751 (cit. on pp. 4, 13, 14).

[IJK09] R. Impagliazzo, R. Jaiswal, and V. Kabanets. "Chernoff-Type Direct Product Theorems". In: *J. Cryptol.* 22.1 (2009), pp. 75–92. DOI: 10.1007/s00145-008-9029-7 (cit. on pp. 4, 13).

[IJKW10] R. Impagliazzo, R. Jaiswal, V. Kabanets, and A. Wigderson. "Uniform Direct Product Theorems: Simplified, Optimized, and Derandomized". In: *SIAM J. Comput.* 39.4 (2010), pp. 1637–1665. DOI: 10.1137/080734030 (cit. on pp. 4, 41).

[Jer23] F. G. Jeronimo. *Fast decoding of explicit almost optimal ε-balanced q-Ary codes and fast approximation of expanding k-CSPs*. en. 2023. DOI: 10.4230/LIPICS.APPROX/RANDOM.2023.60 (cit. on pp. 10, 20, 32, 41).

[Joh62] S. M. Johnson. "A new upper bound for error-correcting codes". en. In: *IEEE transactions on information theory* 8 (3 1962), pp. 203–207. DOI: 10.1109/tit.1962.1057714 (cit. on p. 31).

[JST21] F. G. Jeronimo, S. Srivastava, and M. Tulsiani. "Near-linear time decoding of Ta-Shma's codes via splittable regularity". In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing* (Virtual, Italy). STOC 2021. 2021, pp. 1527–1536. DOI: 10.1145/3406325.3451126 (cit. on pp. 10, 20, 32, 41).

[MWW04] B. D. McKay, N. C. Wormald, and B. Wysocka. "Short cycles in random regular graphs". en. In: *Electronic journal of combinatorics* 11 (1 2004), R66. DOI: 10.37236/1819 (cit. on p. 20).

[Str69] V. Strassen. "Gaussian elimination is not optimal". In: *Numerische mathematik* 13.4 (1969), pp. 354–356 (cit. on p. 1).

[Ta-17] A. Ta-Shma. "Explicit, almost optimal, epsilon-balanced codes". In: *Proceedings of Symposium on Theory of Computing (STOC)* (Montreal, Canada). STOC 2017. 2017, pp. 238–251. DOI: 10.1145/3055399.3055408 (cit. on p. 9).

[TV07] L. Trevisan and S. P. Vadhan. "Pseudorandomness and Average-Case Complexity Via Uniform Reductions". In: *Computational Complexity* 16.4 (2007), pp. 331–364. DOI: 10.1007/s00037-007-0233-x (cit. on pp. 13, 39).

[Val24] G. Valiant. "Matrix Multiplication in Quadratic Time and Energy? Towards a Fine-Grained Energy-Centric Church-Turing Thesis". In: *Proceedings of the Innovations in Theoretical Computer Science Conference (ITCS)*. 2024, 96:1–96:13. DOI: 10.4230/LIPICS.ITCS.2024.96 (cit. on p. 1).

[VD08] V. Volkov and J. Demmel. "Benchmarking GPUs to tune dense linear algebra". In: *Proceedings of the ACM/IEEE Conference on High Performance Computing, SC 2008, November 15-21, 2008, Austin, Texas, USA*. 2008, p. 31. DOI: 10.1109/SC.2008.5214359 (cit. on p. 1).

[VXXZ24] V. Vassilevska Williams, Y. Xu, Z. Xu, and R. Zhou. "New Bounds for Matrix Multiplication: from Alpha to Omega". In: *Proceedings of the Symposium on Discrete Algorithms (SODA)*. 2024, pp. 3792–3835. DOI: 10.1137/1.9781611977912.134 (cit. on p. 1).

[Wor99] N. C. Wormald. "Models of Random Regular Graphs". In: *Surveys in Combinatorics, 1999* (1999), pp. 239–298. DOI: 10.1017/CBO9780511721335.010 (cit. on p. 20).

[ZDCDHSZGQCRZ22] H. Zhou, J. Dong, J. Cheng, W. Dong, C. Huang, Y. Shen, Q. Zhang, M. Gu, C. Qian, H. Chen, Z. Ruan, and X. Zhang. "Photonic matrix multiplication lights up photonic accelerator and beyond". en. In: *Light, science & applications* 11 (1 2022), p. 30. DOI: 10.1038/s41377-022-00717-8 (cit. on p. 1).

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il