Electronic Colloquium on Computational Complexity, Report No. 66 (2025)

An Optimal Error-Correcting Reduction for Matrix Multiplication

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

We present an optimal "worst-case exact to average-case approximate" reduction for matrix multiplication over a finite field of prime order p. Any efficient algorithm that correctly computes, in expectation, at least $(\frac{1}{p} + \varepsilon)$-fraction of entries of the multiplication $A \cdot B$ of a pair (A, B) of uniformly random matrices over the finite field of order p for a positive constant ε can be transformed into an efficient randomized algorithm that computes $A \cdot B$ for all the pairs (A, B) of matrices with high probability. Previously, such reductions were known only in a low-error regime (Gola, Shinkar and Singh; RANDOM 2024) or under non-uniform reductions (Hirahara and Shimizu; STOC 2025).

1 Introduction

Matrix multiplication is a fundamental operation in algebra and plays a central role in scientific computing. Understanding its computational complexity has been a key area of research from both theoretical and practical points of view. Since the pioneering work of Strassen [Str69], numerous studies have been aimed at accelerating matrix multiplication. The best known algorithm currently runs in $O(n^\omega)$ time to multiply two $n \times n$ matrices for some constant $\omega \leq 2.3716$ (see, e.g., [DWZ23; VXXZ24; ADWXXZ25] and references therein), which is significantly faster than the naive $O(n^3)$-time algorithm.

From an applied standpoint, computations are not necessarily limited to traditional computing hardware. Specialized architectures, such as GPUs, enable efficient multiplication of large matrices in practice [VD08]. Additionally, alternative physical models for computing matrix multiplication have been proposed, leveraging optical devices [ZDCDHSZGQCRZ22], thermodynamics [CADMGASCMS23], and even divisible materials like water [Val24]. These approaches would be plagued by a high level of noise and errors, leading to the following natural question:

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

Gola, Shinkar, and Singh [GSS24] recently formulated this question over a finite field $\mathbb{F}$ and gave a solution in a low-error regime. Given a black-box oracle $\mathcal{O}$ that, given a pair of $n \times n$ uniformly random matrices $A, B \sim \mathbb{F}^{n \times n}$, outputs a matrix C such that a $\frac{8}{9}$-fraction of its entries agree with $A \cdot B$ in expectation, they designed a randomized algorithm $M^{\mathcal{O}}$ that can compute matrix multiplication for all inputs with high probability. Very recently, Hirahara and Shimizu [HS25] presented *non-uniform*[1] reductions that improve the fraction $\frac{8}{9}$ of agreement in [GSS24] to

[1] A non-uniform algorithm is an algorithm that takes an advice string α_n for each input length n.

ISSN 1433-8092

$\frac{1}{p} + \varepsilon$ over a finite field of order p for an arbitrary small constant $\varepsilon > 0$, which is optimal. The main open question left in [GSS24; HS25] is whether the optimal fraction $\frac{1}{p} + \varepsilon$ of agreement can be achieved for *uniform* algorithms.

1.1 Our Results

In this work, we answer this question affirmatively. 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 M that outputs the all-0 matrix achieves the expected distance

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

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 *uniform* algorithm.

Theorem 1.1. *Let $\mathbb{F}_p$ be a finite field of prime order p, and let $\varepsilon > 0$ be a constant. Suppose that there exists a randomized algorithm M that runs in time $T(n)$ and satisfies for all sufficiently large n,*

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

where the expectation is taken over uniformly random matrices A, B, and the internal randomness of M. Then, there exists a randomized algorithm M' that runs in time $\widetilde{O}(T(n) + n^2)$ and satisfies, for all sufficiently large n and every $A, B \in \mathbb{F}_p^{n \times n}$,

$$\Pr_{M'} \big[M'(A, B) = AB\big] \geq \frac{2}{3}.$$

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. As an immediate corollary, we obtain the following equivalence.

Corollary 1.2. *The following are equivalent for every prime p and every constant $\varepsilon > 0$.*

- *There exists a randomized $\widetilde{O}(n^2)$-time algorithm M such that for all sufficiently large n,*

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

- *There exists a randomized $\widetilde{O}(n^2)$-time algorithm M' such that for all sufficiently large n,*

$$\mathop{\mathbb{E}}_{\substack{A,B \sim \mathbb{F}_p^{n \times n} \\ M'}} \big[M'(A, B) = AB\big] \geq 1 - o(1).$$

Our results can be interpreted either pessimistically or optimistically. Pessimistically, approximating matrix multiplication slightly better than the trivial algorithm is as difficult as computing matrix multiplication in time $\widetilde{O}(n^2)$, which is the major open question in the long line of research pioneered by Strassen [Str69]. Optimistically, our results give an approach towards resolving this open question: It suffices to design an $\widetilde{O}(n^2)$-time algorithm that correctly computes a *tiny* fraction of entries of the multiplication of a *few* pairs of $n \times n$ matrices![2]

1.2 Proof Overview

We present high-level ideas of the proof of Theorem 1.1. The proof is based on the combination of the techniques developed in the line of research.

1. The result of [GSS24] suggests that it suffices to design an algorithm that computes a $\frac{8}{9}$-fraction of entries of matrix multiplication on average, under the assumption of Theorem 1.1.

2. Following the work of Asadi, Golovnev, Gur, and Shinkar [AGGS22], Hirahara and Shimizu [HS23] presented a "worst-case exact to average-case exact" reduction that transforms any algorithm that exactly computes *all* the entries of matrix multiplication for an α-fraction of matrices for a constant $\alpha > 0$ into a randomized algorithm for matrix multiplication.

Combining these techniques, one can observe that it suffices to design an algorithm that computes a $(1-\delta)$-fraction of entries of matrix multiplication for an α-fraction of matrices for small $\delta, \alpha > 0$ (see the proof of Lemma 3.7 for details).

To achieve this goal, we extend the techniques developed in [HS25] for *uniform* reductions that achieve an optimal fraction of agreement up to a factor of 2. (One of the result in [HS25] is the non-uniform optimal reduction mentioned earlier, whose techniques are orthogonal to ours. The crux of our contributions is to improve uniform but non-optimal reductions of [HS25].) We briefly review the idea of [HS25] and explain the bottleneck of the argument. Let $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ be a linear list-decodable error-correcting code that maps a message of length n to a codeword of length N. Since this is a linear code, there is a matrix $Q \in \mathbb{F}_p^{N \times n}$ such that $\mathsf{Enc}(x) = Qx$. In [HS25], a matrix C is encoded by the *left-right encoding* $\mathsf{Enc}^\star$, which is defined as $\mathsf{Enc}^\star(C) \coloneqq QCQ^\top$.[3] An important property of this encoding is that $\mathsf{Enc}^\star(A \cdot B) = QABQ^\top = (QA) \cdot (QB^\top)^\top = \mathsf{Enc}(A) \cdot \mathsf{Enc}(B^\top)^\top$ for any matrices A and B. This property suggests that if one can approximately compute the multiplication of a pair of matrices $\mathsf{Enc}(A)$ and $\mathsf{Enc}(B^\top)^\top$, then one can recover $A \cdot B$ exactly using a list-decoding algorithm for $\mathsf{Enc}^\star$. Unfortunately, this argument loses a factor of 2: If the list-decoding radius of the original error-correcting code Enc is $1-\beta$, then the list-decoding radius of $\mathsf{Enc}^\star$ can be shown to be at most $1-2\beta$. Even if the original code has the optimal radius of $1 - \frac{1}{p} - \varepsilon$ (i.e., $\beta = \frac{1}{p} + \varepsilon$), the left-right encoding $\mathsf{Enc}^\star$ may not have the optimal list-decoding radius.

To address this issue, we present a new way of approximating $\mathsf{Enc}^\star(A \cdot B)$ by computing matrix multiplication of a pair of "encoded" matrices. We take $\mathsf{Enc}^\star$ to be the derandomized direct sum code of Ta-Shma [Ta-17], which is approximately list-decodable within a radius of $1 - \frac{1}{p} - \varepsilon$ [Jer23]. The derandomized direct sum code is based on a random walk over an expander graph. We design an encoding scheme $\mathsf{Lift}_W(\cdot)$, tailored for this specific code $\mathsf{Enc}^\star$, that satisfies $\mathsf{Enc}^\star(A \cdot B) = \mathsf{Lift}_W(A) \cdot \mathsf{Lift}_W(B^\top)^\top$ (see Definition 3.3, Fig. 1, and Lemma 3.4). Then, $\mathsf{Enc}^\star(A \cdot B)$ can be

[2] In fact, this statement is essentially equivalent to the assumption of Theorem 1.1; see Section A for details.

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

computed by multiplying the pair $(\mathsf{Lift}_W(A), \mathsf{Lift}_W(B^\top)^\top)$ of the encoded matrices, which can be approximated by the algorithm M of the assumption in Theorem 1.1.

One important detail that is omitted in the overview above is that $(\mathsf{Lift}_W(A), \mathsf{Lift}_W(B^\top)^\top)$ may not be uniformly distributed even if A and B are uniformly distributed; thus, M may fail to produce a reasonable approximation of $\mathsf{Lift}_W(A) \cdot \mathsf{Lift}_W(B^\top)^\top$. This issue is addressed in the same way with [HS25] but with a simpler analysis: The matrices can be partitioned into small sub-matrices so that the marginal distribution of each sub-matrix is uniformly distributed. We present a simple greedy coloring procedure that constructs such a partition (see Corollary 3.2), which simplifies the previous construction of the partition in [HS25].

2 Preliminaries

For $n \in \mathbb{N}$, let $[n] = \{1, \dots, n\}$. For a finite set S, we denote by $x \sim S$ that x is chosen uniformly at random from S. For a vector $v \in \mathbb{F}_p^n$ and $i \in [n]$, the i-th entry of v is denoted by v_i. For a subset $I \subseteq [n]$, let $v|_I \in \mathbb{F}_p^{|I|}$ be the restriction of v to I. Formally, for $I = \{i_1, \dots, i_a\} \subseteq [n]$ with $i_1 < \cdots < i_a$, the j-th entry of $v|_I$ is given by v_{i_j} for every $j \in [a]$. We use the same notation for matrices: For a matrix $A \in \mathbb{F}_p^{n \times n}$ and $i, j \in [n]$, the (i,j)-th entry of A is denoted by $A_{i,j}$ and $A|_{I,J} \in \mathbb{F}_p^{|I| \times |J|}$ denotes the restriction of A to $I \times J$.

Error-Correcting Codes and Approximate List-Decoding. Let Σ be a finite set. For two vectors $x, y \in \Sigma^n$, let $\mathrm{dist}(x,y) = \frac{1}{n} \sum_{i \in [n]} \mathbf{1}_{x(i) \neq y(i)}$ be the normalized Hamming distance between x and y. For a vector $x \in \Sigma^n$ and $\rho \in [0,1]$, let $\mathrm{Ball}(x, \rho) \subseteq \Sigma^n$ be the set of vectors $y \in \Sigma^n$ that satisfy $\mathrm{dist}(x,y) \leq \rho$. A *code* $\mathcal{C} \subseteq \Sigma^m$ is a subset of Σ^m. The *block length* of $\mathcal{C}$ is m. An element of a code $\mathcal{C}$ is called a *codeword*. In this paper, we prefer to use the term *code* to refer to an *encoding function*, a function $\mathsf{Enc}\colon \Sigma^n \to \Sigma^m$ that maps a message $x \in \Sigma^n$ to a codeword $\mathsf{Enc}(x) \in \Sigma^m$. The finite set Σ is usually a finite field $\mathbb{F}_p$.

We review the notion of approximate list-decoding, which has been studied in the literature of direct product encoding and hardness amplification [DHKNT21; IJKW10].

Definition 2.1 (approximate list-decoding)**.** *A code* $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^N$ *is* δ-approximately 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.

In particular, the case of $\delta = 0$ corresponds to the standard list-decodability.

Expander Graphs. The *girth* of a graph G is the minimum length of cycles in G. Note that the girth of G is at least g if and only if G does not contain any cycle of length less than g. For a d-regular 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 (if G is simple, $m_{u,v} \in \{0,1\}$). 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 satisfy $\max\{|\lambda_2|, |\lambda_{|V|}|\} \leq \lambda$.

We use a sparse regular expander with large girth as a building block of our reduction. For example, we can use a random d-regular graph $G_{n,d}$ for a sufficiently large $d = O(1)$, which satisfies the following properties.

Lemma 2.2. *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 hold:*

- *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].*[4]
- *There exists a constant $c > 0$ that depends only on d such that with probability $1 - n^{-10}$, $G_{n,d}$ is a $\left(\frac{2\sqrt{d-1}}{d} + c\left(\frac{\log\log n}{\log n}\right)^2\right)$-expander [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].*

Expander-Walk Direct Sum Codes. Our key technical tool is an approximate list-decoding algorithm for direct sum encoding with respect to a collection of k-walks on a regular expander graph. Jeronimo, Srivastava, and Tulsiani [JST21] and Jeronimo [Jer23] implicitly presented a randomized approximate list-decoding algorithm for direct sum encoding with respect to a *splittable* collection $W \subseteq [n]^k$ of k tuples, which satisfies a certain expansion property. A canonical example of a splittable collection is the collection of walks on a regular expander graph. To state it more formally, a *k-walk* on a graph $G = ([n], E)$ is a k-tuple $w = (u_1, \ldots, u_k) \in [n]^k$ of vertices such that $\{u_i, u_{i+1}\} \in E$ for all $i \in [k-1]$. The set W of all possible k-walks is referred to as the *k-walk collection*. For a walk $w = (u_1, \ldots, u_k) \in W$, let $\mathsf{visit}(w) = \{u_1, \ldots, u_k\}$ be the set of vertices visited by w. If G is d-regular for a constant $d \in \mathbb{N}$ and $k = O(1)$, then $|W| = nd^{k-1} = O(n)$.

Definition 2.3 (Expander-Walk Direct Sum Code)**.** *Let $G = ([n], E)$ be a d-regular graph and $W \subseteq [n]^k$ be the k-walk collection on G. The* k-wise expander walk direct sum code *with respect to G is the code $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^W$ that maps $x \in \mathbb{F}_p^n$ to $\mathsf{Enc}(x) \in \mathbb{F}_p^W$ defined by*

$$\mathsf{Enc}(x)_w = x_{u_1} + \cdots + x_{u_k}$$

for every $w = (u_1, \ldots, u_k) \in W$.

Lemma 2.4 (Implicit in the proof of [Jer23, Theorem 7.5])**.** *Let $p \geq 2$ be a constant prime, $\beta, \gamma, \lambda > 0$ and $k \in \mathbb{N}$ be any constants such that*

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

Then, for some $\ell = O(1)$, the k-wise expander-walk direct sum code Enc is γ-approximately ℓ-list-decodable within radius $(1-1/p)(1-\beta)$ by an $O(|W|\mathsf{polylog}(|W|))$-time randomized algorithm.

A brief description of the decoding algorithm of Lemma 2.4 can be found in Section B.

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

Auxiliary Results. We present three auxiliary results that are used in the proof of Theorem 1.1. First, we show that multiplying random *rectangular* matrices can be reduced to multiplying two random *square* matrices.

Lemma 2.5 ([HS25, 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,*

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

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

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

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 $I=\{i_1,\dots,i_\ell\}\subseteq[n]$ 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 $\mathcal{O}(\overline{A},\overline{B})|_{I,J}$.

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

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

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

In the following, we formally state the result of Gola, Shinkar, and Singh [GSS24], which is based on a simple and elegant modification of the worst-case to average-case reduction of Blum, Luby, and Rubinfeld [BLR93].

Lemma 2.6 (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*

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

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

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

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

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

Lemma 2.7. *For any $\alpha,\beta>0$, there exists an $O(n^2+n\log n/\beta^2)$-time randomized algorithm M that, on input $A,B,C\in\mathbb{F}_p^{n\times n}$, satisfies the following:*

- *If $\mathrm{dist}(AB,C)\le\alpha$, then M accepts with probability $1-1/n^3$.*
- *If $\mathrm{dist}(AB,C)\ge\alpha+\beta$, then M rejects 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 $(AB)_{i,j}=C_{i,j}$ for uniformly random $i,j\sim[n]$ for $T=O\left(\frac{\log n}{\beta^2}\right)$ times. The algorithm accepts if and only if the number of iterations at which $(AB)_{i,j}=C_{i,j}$ holds is at least $(1-\alpha-\beta/2)\cdot T$.

If $\mathrm{dist}(AB,C)\le\alpha$, then $(AB)_{i,j}=C_{i,j}$ with probability at least $1-\alpha$ over the random choice of $i,j\sim[n]$. Therefore, by the Hoeffding bound, the algorithm M' accepts with probability $1-\exp(-2\beta^2T)\ge 1-1/n^3$. The case of $\mathrm{dist}(AB,C)\ge\alpha+\beta$ is the same. □

Finally, we will use the following direct product lemma.

Lemma 2.8 (Direct product lemma; [IJK09; HS23; HS24]). *Let D be a set. For all sufficiently small $\delta^*,\varepsilon^*>0$, for every $K\ge O(\log(1/\varepsilon^*))/(\delta^*\varepsilon^*)^2$, for every function $S\colon D^K\to[0,1]$, it holds that*

$$\Pr_{x\sim D}\left[\left|\mathop{\mathbb{E}}_{y'\sim\Gamma(x)}\left[S(y')\right]-\mathop{\mathbb{E}}_{y\sim\mathcal{D}^K}[S(y)]\right|\le\varepsilon^*\right]\ge 1-\delta^*.$$

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'.

3 Optimal Reduction for Small Field

In this section, we prove Theorem 1.1.

3.1 Vertex-Disjoint Partition of k-Walks

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. A subset $P \subseteq W$ of the k-walk collection is said to be *vertex-disjoint* if walks in P visit distinct vertices, that is, $\mathsf{visit}(w) \cap \mathsf{visit}(w') = \emptyset$ for any distinct pair of walks $w, w' \in P$.

Lemma 3.1. *Let $k, d \in \mathbb{N}$ be constants, $G = ([n], E)$ 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 $P_1 \sqcup \cdots \sqcup P_a = W$ of W such that, for every $i \in [a]$, the set P_i is vertex-disjoint. Moreover, we can compute such a partition in time $O(|W|)$.*

Proof. For the k-walk collection W, define the intersection graph $I = (V(I), E(I))$ with respect to W as follows.

- The vertex set $V(I)$ is defined to be W.
- The edge set $E(I)$ is defined to be the set of all the pairs (w, w') of distinct walks in W such that $\mathsf{visit}(w) \cap \mathsf{visit}(w') \neq \emptyset$.

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 $\sum_{i=0}^{k-1} d^i \cdot d^{k-i-1} = k \cdot d^{k-1}$ walks in W, the degree of the I-vertex w on I is at most $k \cdot kd^{k-1}$. That is, the maximum degree of I is at most $k^2 d^{k-1}$. It follows that 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, no pair of vertices in P_i is connected by an edge in I). In other words, each P_i is vertex-disjoint. □

Corollary 3.2. *Let $k, d \in \mathbb{N}$ be constants, $G = ([n], E)$ be a d-regular graph with girth at least k, and $W \subseteq [n]^k$ be the k-walk collection on G. Then, there exist an integer $a = a(k, d) \in \mathbb{N}$ and a partition $\{P_1, \ldots, P_a, P'\}$ of $W = P_1 \sqcup \cdots \sqcup P_a \sqcup P'$ with the following properties.*

- *P_i is vertex-disjoint for every $i \in [a]$.*
- *$|P_i| \leq n/k$ for every $i \in [a]$.*
- *Every walk w in P_i visits exactly k vertices (i.e., $|\mathsf{visit}(w)| = k$) for every $i \in [a]$.*
- *$|P'| \leq \frac{k}{d} \cdot |W|$.*

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

Proof. Let $P'_1 \sqcup \cdots \sqcup P'_a = W$ be the partition of Lemma 3.1. For every $i \in [a]$ let $P_i \subseteq P'_i$ be the set of walks $w \in P'_i$ such that $|\mathsf{visit}(w)| = k$. Define $P' = W \setminus (P_1 \cup \cdots \cup P_a)$.

It is easy to observe that the first three properties are satisfied. Since P_i is vertex-disjoint and every walk in P_i visits exactly k vertices, P_i contains at most n/k walks. To prove the last property, 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 $j \in \{2, \ldots, k-1\}$. By the union bound over $j \in \{2, \ldots, k-1\}$, 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

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

□

3.2 Worst-Case Exact to Average-Case Approximate Reduction

In this section, we show that an average-case approximation solver can be used to compute all entries of the product of any given two matrices, proving Theorem 1.1.

Definition 3.3 (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 $\mathsf{Lift}_W(A)$ is given by $(a_{i_s})_t$.*

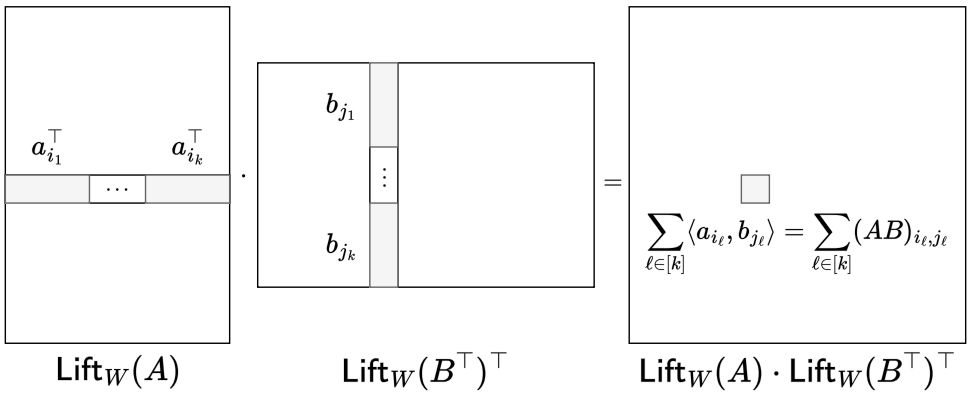

Figure 1: The $(\mathbf{i}, \mathbf{j})$-th entry of $\mathsf{Lift}_W(A) \cdot \mathsf{Lift}_W(B)$ for $\mathbf{i} = (i_1, \ldots, i_k)$ and $\mathbf{j} = (j_1, \ldots, j_k)$ is equal to $\sum_{\ell=1}^k (A \cdot B)_{i_\ell, j_\ell}$.

In the following lemma, we show a useful relation between W-lifting and direct sum encoding. See Fig. 1 to gain some intuition. To state the lemma, recall that the tensor product $G \otimes G$ for a graph $G = ([n], E)$ is the graph on vertex set $[n] \times [n]$ such that two vertices $(u_1, u_2), (v_1, v_2) \in [n] \times [n]$ are adjacent on $G \otimes G$ if and only if $\{u_i, v_i\} \in E$ for both $i = 1, 2$. It is well known that, if G is d-regular and λ-expander, then $G \otimes G$ is d^2-regular and λ-expander.[5]

By definition, if $\mathbf{i} = (i_1, \ldots, i_k), \mathbf{j} = (j_1, \ldots, j_k)$ are k-walks on G, then the sequence of pairs $(i_1, j_1), \ldots, (i_k, j_k)$ forms a k-walk on $G \otimes G$. With this in mind, the expander-walk direct sum encoding on $G \otimes G$ (Definition 2.3) can be naturally represented as a function $\mathsf{Enc}\colon \mathbb{F}_p^{n \times n} \to \mathbb{F}_p^{W \times W}$

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

that maps a message matrix $C \in \mathbb{F}_p^{n\times n}$ to a codeword matrix $\mathsf{Enc}(C) \in \mathbb{F}_p^{W\times W}$ such that the $(\mathbf{i},\mathbf{j})$-th entry of $\mathsf{Enc}(C)$ is given by $\sum_{\ell=1}^{k} C_{i_\ell,j_\ell}$. Note that $\mathsf{Enc}(C) \in \mathbb{F}_p^{W\times W}$ can be identified with an $nd^{k-1} \times nd^{k-1}$ matrix over $\mathbb{F}_p$.

Lemma 3.4. *Let $G = ([n], E)$ be a d-regular graph and $W \subseteq [n]^k$ be the k-walk collection. 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*

$$\mathsf{Lift}_W(A) \cdot \mathsf{Lift}_W(B^\top)^\top = \mathsf{Enc}(AB).$$

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

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

□

Using Lemmas 3.1 and 3.4, we show that an average-case approximation solver can be used to compute an approximate codeword of the direct sum encoding.

Lemma 3.5. *Let $\mathbb{F}_p$ be the finite field of order p, and let $k, d \in \mathbb{N}$ and $\alpha > 0$ be constants. There exists a randomized oracle algorithm M such that, for every k, every d-regular graph $G = ([n], E)$ whose girth is at least k, and every oracle $\mathcal{O}$ such that for all sufficiently large n,*

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

it holds that for all sufficiently large n,

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

The algorithm M runs in time $O(n^2)$ and makes $O(1)$ queries.

Proof. The oracle algorithm $M^{\mathcal{O}}$ operates on 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 $W = P_1 \sqcup \cdots \sqcup P_a \sqcup P'$ of Corollary 3.2.

3. For each $i, j \in [a]$, run the algorithm of Lemma 2.5 on input $\left(\mathsf{Lift}_W(A)|_{P_i,[kn]}, \mathsf{Lift}_W(B^\top)^\top|_{[kn],P_j}\right)$ using $\mathcal{O}$ as oracle. Let $\widetilde{C}^{(i,j)} \in \mathbb{F}_p^{|P_i|\times|P_j|}$ be the output of this algorithm.

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') arbitrarily.

Note that M runs in time $O(n^2)$ and makes at most $a^2 = O(1)$ queries.

We prove Eq. (2). For each $i \in [a]$, the set P_i consists of vertex-disjoint walks w such that $|\mathsf{visit}(w)| = k$. Since the $\mathbf{i}$-th row vector of A consists of i_1-th, i_2-th, $\ldots$, i_k-th row vectors of A for every $\mathbf{i} = (i_1, \ldots, i_k)$ and vertices visited by walks in P_i are distinct, 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 $\left(\mathsf{Lift}_W(A)|_{P_i,[kn]}, \mathsf{Lift}_W(B^\top)^\top|_{[kn],P_j}\right)$ is uniform over $\mathbb{F}_p^{|P_i|\times kn} \times \mathbb{F}_p^{kn\times|P_j|}$. Since $|P_i|, |P_j| \le kn$ for each $i, j \in [a]$, it follows from Lemma 2.5 that

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

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 2.5.

Therefore, we have

$$\begin{aligned}
\text{LHS of Eq. (2)} &= \mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M^{\mathcal{O}}}}\left[\mathrm{dist}\left(\widetilde{C},\, \mathsf{Lift}_W(A)\cdot\mathsf{Lift}_W(B^\top)^\top\right)\right] && \because \text{Lemma 3.4}\\
&\le \mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M^{\mathcal{O}}\\ i,j\sim[a]}}\left[\mathrm{dist}\left(\widetilde{C}^{(i,j)},\, \mathsf{Lift}_W(A)|_{P_i,[kn]}\cdot\mathsf{Lift}_W(B^\top)^\top|_{[kn],P_j}\right)\right] + \frac{2|P'|}{|W|}\\
&\le 1 - \alpha + \frac{2k}{d}.
\end{aligned}$$

□

Now we combine the above lemma with the list-decoding algorithm of Lemma 2.4 to compute a $(1-\delta)$-fraction of the entries of AB for an $\Omega(\varepsilon)$-fraction of pairs (A, B) of matrices.

Lemma 3.6. *Let $\mathbb{F}_p$ be a finite field of prime order p and $\delta, \varepsilon > 0$ be constants. There exists a randomized $O(1)$-query oracle algorithm $M^{\mathcal{O}}$ that runs in time $\widetilde{O}(n^2)$ such that, for any oracle $\mathcal{O}$ satisfying*

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

it holds for every sufficiently large n that

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

Proof. Fix a randomized oracle $\mathcal{O}$. For given $p \geq 2$, $\delta, \varepsilon > 0$, we set constants $k \in \mathbb{N}$ and $\lambda > 0$ such that Eq. (1) holds for $\beta = \varepsilon/4$ and $\gamma = \delta/2$. Note that we can set $k = O(p^4 \log(1/\varepsilon)/\delta^2)$ and $\lambda = O(\varepsilon^2/k^3)$. For $d = \lceil \max\{\frac{4}{\lambda^2}, \frac{8k}{\varepsilon}\} \rceil$, fix a d-regular λ-expander graph $G = ([n], E)$ with girth at least k. From Lemma 2.2, with probability $\Omega(1)$ a random regular graph $G_{n,d}$ satisfies this property. We repeat generating $G = G_{n,d}$ until G has girth at least k, which can be checked by computing the girth in time $O(n^2)$ by the breadth first search from all vertices. Note that $O(\log n)$ samples from $G_{n,d}$ suffice to obtain such G with probability $1 - n^{-\Omega(1)}$. Thus, we can construct such G in time $\widetilde{O}(n^2)$ with probability $1 - o(1)$.

From Lemma 3.5 for $\alpha = \frac{1}{p} + \varepsilon$, there exists a randomized oracle algorithm $M_0^{\mathcal{O}}$ such that

$$\begin{aligned}\mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M_0^{\mathcal{O}}}}\left[\operatorname{dist}\left(M_0^{\mathcal{O}}(G,k,A,B),\, \mathsf{Enc}(AB)\right)\right] &\leq 1 - \frac{1}{p} - \varepsilon + \frac{2k}{d} \\ &\leq 1 - \frac{1}{p} - \frac{3\varepsilon}{4}. \qquad \because\ d \geq 8k/\varepsilon\end{aligned}$$

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

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

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

We present an algorithm that outputs a matrix that is $(1-\delta)$-close to AB 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 $\operatorname{dist}(\widetilde{C}, \mathsf{Enc}(AB))) \leq 1 - \frac{1}{p} - \frac{\varepsilon}{4}$. 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 2.4, this encoding Enc is $\delta/2$-approximately ℓ-list-decodable within radius $\left(1 - \frac{1}{p}\right)(1-\beta) \leq 1 - \frac{1}{p} - \frac{\varepsilon}{4}$ by an $\widetilde{O}(n^2)$-time algorithm. Thus, by running the list-decoding algorithm on input $\widetilde{C} \in \mathbb{F}_p^{W\times W}$, we obtain a set L of $O(1)$ matrices that contains a matrix that is $\delta/2$-close to AB in time $\widetilde{O}(n^2)$. Finally, output a matrix in the list L such that the the verifier of Lemma 2.7 with $\alpha = \beta = \delta/2$ accepts. □

We then show that we can indeed compute a good approximation for a $(1 - O(\delta))$-fraction of (A, B) (Lemma 3.7).

Lemma 3.7. *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*

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

it holds for every sufficiently large n that

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

Proof. From Lemmas 2.5 and 3.6, for any constant $\delta_0 > 0$ and any $\ell \le n$, there exists an oracle algorithm $M_0^{\mathcal{O}}$ such that

$$\Pr_{\substack{A\sim\mathbb{F}_p^{\ell\times n}\\ B\sim\mathbb{F}_p^{n\times\ell}}}\left[\Pr_{M_0^{\mathcal{O}}}[\mathrm{dist}\big(M_0^{\mathcal{O}}(A,B),\, AB\big) \le \delta_0] \ge 1 - o(1)\right] \ge \frac{\varepsilon}{4}. \tag{3}$$

For a sufficiently large constant $c > 0$, define the following parameters:

$$K = \frac{c\log(1/\varepsilon)}{\gamma^2\varepsilon^2}, \quad \delta = \frac{\gamma}{4K^2}, \text{ and } m = \left\lfloor \frac{n}{K} \right\rfloor.$$

Let $M_0^{\mathcal{O}}$ be the algorithm of Eq. (3) for $\delta_0 = \delta/2$ and $\ell = Km \le n$. Let $I_1 \sqcup \cdots \sqcup I_K \sqcup I'$ be the partition of $[n]$ defined by

$$I_1 = \{1,\dots,m\},\ I_2 = \{m+1,\dots,2m\},\dots,\ I_K = \{(K-1)m+1,\dots,Km\},\ \text{and } I' = \{Km+1,\dots,n\}.$$

For $i, j \in [K]$, let $A_i = A|_{I_i,[n]}$ and $B_j = B|_{[n],I_j}$ (Fig. 2).

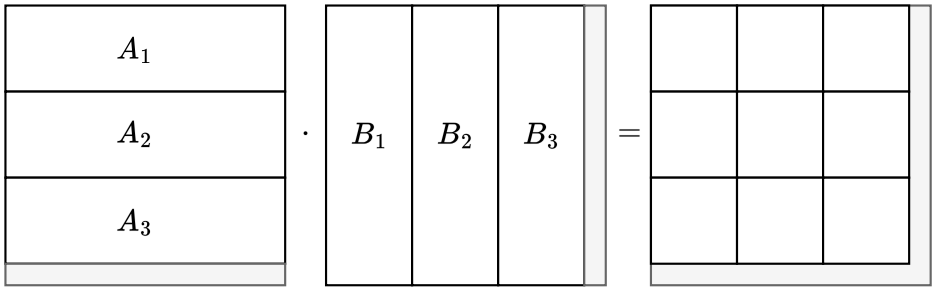

Figure 2: Partition of A, B into $A_1,\dots,A_K$ and $B_1,\dots,B_K$ considered in the proof of Lemma 3.7. To compute an approximation of AB, it suffices to compute a good approximation of A_iB_j for all i, j.

Our reduction $M^{\mathcal{O}}$ is given $A, B \in \mathbb{F}_p^{n\times n}$ as input and runs 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 $i, j \in [K]$, repeat the following for $O(\log(1/\gamma)/\varepsilon)$ times:
   (a) Sample $\overline{A} \sim \mathbb{F}_p^{Km\times n}, \overline{B} \sim \mathbb{F}_p^{n\times Km}$ and $k \sim [K]$.
   (b) Replace $\overline{A}|_{I_k,[n]}$ by A_i. Similarly, replace $\overline{B}|_{[n],I_k}$ by B_j (Fig. 3).
   (c) Run the oracle algorithm $M_0^{\mathcal{O}}(\overline{A},\overline{B})$. Let $C \in \mathbb{F}_p^{Km\times Km}$ be the output.
   (d) If $C|_{I_i,I_j}$ is δ-close to A_iB_j (which can be checked by Lemma 2.7), replace $D|_{I_i,I_j}$ by $C|_{I_i,I_j}$.
3. Output D.

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

$$\begin{aligned}
\mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M^{\mathcal{O}}}}\left[\mathrm{dist}\big(M^{\mathcal{O}}(A,B),\, AB\big)\right] &\le \mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M^{\mathcal{O}}\\ i,j\sim[K]}}\left[\mathrm{dist}\big(M^{\mathcal{O}}(A,B)|_{I_i,I_j},\, (AB)|_{I_i,I_j}\big)\right] + \frac{2|I'|}{n} \\
&\le \mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M^{\mathcal{O}}\\ i,j\sim[K]}}\left[\mathrm{dist}\big(M^{\mathcal{O}}(A,B)|_{I_i,I_j},\, A_iB_j\big)\right] + \frac{2m}{n} \\
&\le \mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M^{\mathcal{O}}\\ i,j\sim[K]}}\left[\mathrm{dist}\big(M^{\mathcal{O}}(A,B)|_{I_i,I_j},\, A_iB_j\big)\right] + \frac{\gamma}{4}. \qquad (4)
\end{aligned}$$

In the last inequality, we used $m/n \le 1/K \le \gamma/8$ by our choice of K (the leading constant c in the definition of K is sufficiently large but chosen independently of γ and ε since $K \ge c/\gamma^2$). To bound Eq. (4), we shall take a closer look at $\mathbb{E}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M^{\mathcal{O}}}}\left[\mathrm{dist}\big(M^{\mathcal{O}}(A,B)|_{I_i,I_j},\, A_iB_j\big)\right]$ for every $i,j\in[K]$.

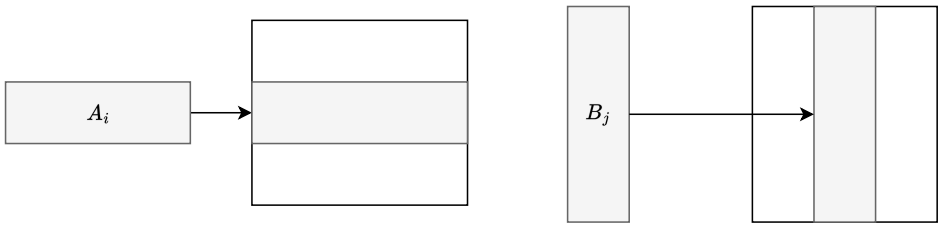

Figure 3: The query $(\overline{A},\overline{B})$ of the oracle algorithm $M^{\mathcal{O}}$ is obtained by embedding A_i and B_j into $\overline{A}$ and $\overline{B}$ at random positions. This procedure of generating a query coincides with the sampling procedure $\Gamma(x)$ of Lemma 2.8 for $x=(A_i,B_j)\sim\mathbb{F}_p^{m\times n}\times\mathbb{F}_p^{n\times m}$

Call an instance $(\overline{A},\overline{B})\in\mathbb{F}_p^{Km\times n}\times\mathbb{F}_p^{n\times Km}$ *good* if $\mathbb{E}_{M_0^{\mathcal{O}}}\left[\mathrm{dist}\big(M_0^{\mathcal{O}}(\overline{A},\overline{B}),\, \overline{A}\cdot\overline{B}\big)\right]\le\delta$. From Eq. (3) with $\ell=Km$ and $\delta_0=\delta/2$, at least an $\varepsilon/4$-fraction of $(\overline{A},\overline{B})$ are good. Let $C\in\mathbb{F}_p^{Km\times Km}$ be the matrix obtained at Step 2(c) at an iteration where $i=j=1$ (other blocks can be shown similarly). For $x=(A_1,B_1)$, the query $y=(\overline{A},\overline{B})$ is obtained by embedding A_1 and B_1 into $\overline{A}$ and $\overline{B}$ at random positions (Fig. 3). This coincides with the sampling procedure $\Gamma(x)$ of Lemma 2.8. Therefore, from Lemma 2.8 for S being the indicator of the set of good instances, $\delta^*=\gamma/4$ and $\varepsilon^*=\varepsilon/8$, for at least a $(1-\gamma/4)$-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 $i=j=1$), the probability that at least one instance $(\overline{A},\overline{B})$ is good (and thus the matrix C is δ-close to AB) is at least $1-\gamma/4$. If C is δ-close to $\overline{A}\cdot\overline{B}$, the number of disagreeing entries between C and A_1B_1 is at most $\delta\cdot(Km)^2$; thus the submatrix $C|_{I_1,J_1}$ agrees with A_1B_1 on at least $m^2-\delta(Km)^2=(1-\delta K^2)m^2$ entries, that is, $\mathrm{dist}(C|_{I_1,J_1},\, A_1B_1)\le\delta K^2\le\frac{\gamma}{4}$

by our choice of δ. Therefore, we have

$$\begin{aligned}(4) &= \mathop{\mathbb{E}}_{\substack{(A_1,B_1)\sim\mathbb{F}_p^{m\times n}\times\mathbb{F}_p^{n\times m}\\ \overline{A},\overline{B}\\ C=M_0^{\mathcal{O}}(\overline{A},\overline{B})}} [\mathrm{dist}(C|_{I_1,J_1}, A_1B_1)] + \frac{\gamma}{4} \\ &\leq \underbrace{\frac{\gamma}{4}}_{\text{sampler property}} + \underbrace{\frac{\gamma}{4}}_{\substack{\text{no } (\overline{A},\overline{B}) \text{ is good}\\ \text{during repetition}}} + \underbrace{\frac{\gamma}{4}}_{\mathrm{dist}\left(C|_{I_1,J_1}, A_1B_1\right)} + \frac{\gamma}{4} \\ &\leq \gamma.\end{aligned}$$

□

Proof of Theorem 1.1. From Lemma 3.7 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 Lemma 2.6, we obtain the algorithm as desired. □

References

[ADWXXZ25] Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei 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. URL: https://epubs.siam.org/doi/10.1137/1.9781611978322.63 (cit. on p. 1).

[AGGS22] Vahid R. Asadi, Alexander Golovnev, Tom Gur, and Igor 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 p. 3).

[BLR93] Manuel Blum, Michael Luby, and Ronitt 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 p. 6).

[Bor15] Charles 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). URL: http://arxiv.org/abs/1502.04482 (visited on 11/02/2024) (cit. on p. 5).

[CADMGASCMS23] Patrick J. Coles, Maxwell Aifer, Kaelan Donatella, Denis Melanson, Max Hunter Gordon, Thomas Dybdahl Ahle, Daniel Simpson, Gavin Crooks, Antonio J Martinez, and Faris Mouti Sbahi. "Thermodynamic AI and Thermodynamic Linear Algebra". In: *Machine Learning with New Compute Paradigms*. 2023 (cit. on p. 1).

[DHKNT21] Irit Dinur, Prahladh Harsha, Tali Kaufman, Inbal Livni Navon, and Amnon Ta-Shma. "List-Decoding with Double Samplers". In: *SIAM Journal on Computing* 50 (2 2021), pp. 301–349. DOI: 10.1137/19M1276650. URL: https://doi.org/10.1137/19M1276650 (cit. on p. 4).

[DWZ23] Ran Duan, Hongxun Wu, and Renfei 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).

[Fri03] Joel 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. URL: https://doi.org/10.1145/780542.780646 (visited on 04/24/2024) (cit. on p. 5).

[GGR11] Parikshit Gopalan, Venkatesan Guruswami, and Prasad 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. 3).

[GSS24] Ashish Gola, Igor Shinkar, and Harsimran 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. 1–3, 6, 7).

[HJ91] Roger A Horn and Charles R Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991. DOI: 10.1017/CBO9780511840371. (Visited on 10/16/2024) (cit. on p. 9).

[HS23] Shuichi Hirahara and Nobutaka 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. 3, 7).

[HS24] Shuichi Hirahara and Nobutaka 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 p. 7).

[HS25] Shuichi Hirahara and Nobutaka Shimizu. "Error-Correction of Matrix Multiplication Algorithms". In: *Proceedings of the Annual ACM SIGACT Symposium on Theory of Computing (STOC 2025)*. 2025 (cit. on pp. 1–4, 6).

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

[IJKW10] Russell Impagliazzo, Ragesh Jaiswal, Valentine Kabanets, and Avi 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 p. 4).

[Jer23] Fernando 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. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.APPROX/RANDOM.2023.60 (cit. on pp. 3, 5, 19).

[JST21] Fernando G. Jeronimo, Shashank Srivastava, and Madhur 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. URL: https://doi.org/10.1145/3406325.3451126 (cit. on pp. 5, 19).

[MWW04] Brendan D. McKay, Nicholas C. Wormald, and Beata Wysocka. "Short cycles in random regular graphs". en. In: *Electronic journal of combinatorics* 11 (1 2004), R66. DOI: 10.37236/1819. URL: https://www.combinatorics.org/ojs/index.php/eljc/article/view/v11i1r66 (visited on 11/03/2024) (cit. on p. 5).

[Str69] Volker Strassen. "Gaussian elimination is not optimal". In: *Numerische mathematik* 13.4 (1969), pp. 354–356. DOI: 10.1007/BF02165411 (cit. on pp. 1, 3).

[Ta-17] Amnon 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. URL: https://doi.org/10.1145/3055399.3055408 (visited on 09/09/2022) (cit. on p. 3).

[Val24] Gregory 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] Vasily Volkov and James 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] Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei 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] Nicholas C. Wormald. "Models of Random Regular Graphs". In: *Surveys in Combinatorics, 1999* (1999), pp. 239–298. DOI: 10.1017/CBO9780511721335.010. URL: http://dx.doi.org/10.1017/CBO9780511721335.010 (cit. on p. 5).

[ZDCDHSZGQCRZ22] Hailong Zhou, Jianji Dong, Junwei Cheng, Wenchan Dong, Chaoran Huang, Yichen Shen, Qiming Zhang, Min Gu, Chao Qian, Hongsheng Chen, Zhichao Ruan, and Xinliang 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. URL: https://www.nature.com/articles/s41377-022-00717-8 (cit. on p. 1).

A Average-Case Approximation vs. Expected Distance

We show that computing a small fraction of entries of AB for a small fraction of $(A, B) \sim \mathbb{F}_p^{n\times n} \times \mathbb{F}_p^{n\times n}$ suffices to meet the requirement of Theorem 1.1. Specifically, we prove the following:

Lemma A.1. *Let $\mathbb{F}_p$ be a finite field of order p, and let $\varepsilon > 0$ be a constant. Suppose there exists an algorithm M that runs in time $T(n)$ and satisfies for all sufficiently large n,*

$$\Pr_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M}}\left[\operatorname{dist}(M(A,B),AB) \le 1-\frac{1}{p}-\varepsilon\right] \ge \varepsilon.$$

Then, there exists an algorithm M' that runs in time $\widetilde{O}(T(n)+n^2)$ and satisfies for all sufficiently large n,

$$\mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M'}}\left[\operatorname{dist}(M'(A,B),AB)\right] \le 1-\frac{1}{p}-\Omega\big(\varepsilon^2\big).$$

Conversely, suppose there exists an algorithm L that runs in time $T(n)$ and satisfies for all sufficiently large n,

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

Then, the same algorithm L satisfies

$$\Pr_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ L}}\left[\operatorname{dist}(L(A,B),AB) \le 1-\frac{1}{p}-\frac{\varepsilon}{2}\right] \ge \frac{\varepsilon}{2}.$$

Proof. Let M' be the algorithm that, on input (A, B), runs $C := M(A, B)$ and outputs C if the verification algorithm of Lemma 2.7 on input A, B, C accepts for $\alpha = \beta = \varepsilon$ and outputs a random matrix otherwise.

With probability at least $\varepsilon - o(1)$ (over the choice of A, B and the internal randomness of M), the algorithm M' outputs a matrix $C = M(A, B)$ such that $\operatorname{dist}(C, AB) \le 1 - \frac{1}{p} - \varepsilon$. With probability at most $1 - \varepsilon + o(1)$, the algorithm M' outputs a random matrix, which satisfies $\mathsf{E}_{M'}[\operatorname{dist}(M'(A, B), AB)] = 1-\frac{1}{p}$. Therefore, the expected distance of $M'(A, B)$ and AB is bounded by

$$\begin{aligned}\mathop{\mathbb{E}}_{\substack{A,B\sim\mathbb{F}_p^{n\times n}\\ M'}}\left[\operatorname{dist}(M'(A,B),AB)\right] &\le \varepsilon\left(1-\frac{1}{p}-\varepsilon\right)+(1-\varepsilon)\left(1-\frac{1}{p}\right)+o(1)\\ &= 1-\frac{1}{p}-\Omega(\varepsilon^2)\end{aligned}$$

for sufficiently large n.

The "Conversely" part follows from Markov's inequality. Note that

$$\Pr_{A,B,L}\left[\mathrm{dist}(L(A,B),AB) \geq 1-\frac{1}{p}-\frac{\varepsilon}{2}\right] \leq \frac{1-1/p-\varepsilon}{1-1/p-\varepsilon/2} \leq 1-\frac{\varepsilon}{2}.$$

□

B Approximate List-Decoding

For completeness, we present a proof sketch of Lemma 2.4 by [Jer23]. Since the decoding algorithm and its analysis rely on the heavy machinery of regularity lemma, we omit the technical details and just describe the decoding algorithm.

Proof Sketch of Lemma 2.4. Let $\mathsf{Enc}\colon \mathbb{F}_p^n \to \mathbb{F}_p^W$ be the k-wise expander-walk direct sum code, where $W \subseteq [n]^k$ is the k-walk collection on a d-regular λ-expander graph $G = ([n], E)$. For a partition $\mathcal{P} = \{P_1, \dots, P_a\}$ of $[n]$, we say that a vector $x \in \mathbb{F}_p^n$ is *$\mathcal{P}$-measurable* if for every $P_i \in \mathcal{P}$, all entries of the restriction $x|_{P_i}$ are the same. Note that the number of $\mathcal{P}$-measurable vectors is q^a.

Our decoding algorithm is a slight modification of Algorithm 7.7 of [Jer23]. On input $\tilde{y} \in \mathbb{F}_p^W$, the decoding algorithm runs as follows:

1. Let $L = \emptyset$.
2. For every $a \in \mathbb{F}_p \setminus \{0\}$, do the following:
   (a) Let $g^{(a)} \in \mathbb{C}^{n^k}$ be the vector defined by
   $$g^{(a)}_{i_1,\dots,i_k} = \begin{cases} \omega^{a\cdot\tilde{y}_{(i_1,\dots,i_k)}} & \text{if } (i_1,\dots,i_k) \in W, \\ 0 & \text{otherwise,} \end{cases}$$
   where $\omega = \exp(-2\pi\mathrm{i}/p)$ is the primitive p-th root of unity (note that p is a prime).
   (b) Compute a partition $\mathcal{P}^{(a)}$ of $[n]$ given oracle access to $g^{(a)}$ using the efficient weak regularity lemma [Jer23, Theorem 5.12]. This can be done in time $O(|W|) = O(n)$ and the number of subsets in $\mathcal{P}^{(a)}$ can be shown to be $O(1)$ (here, we used the assumption that G is an expander).
   (c) Add all $\mathcal{P}^{(a)}$-measurable vectors to L.
3. Output L.

Obviously, the output L contains at most $O(1)$ vectors in $\mathbb{F}_p^n$. The proof of the correctness directly follows from the analysis of Algorithm 7.7 of [Jer23]. □

Indeed, [Jer23; JST21] considers the code obtained by applying the distance amplification to a unique-decodable code using direct sum over a splittable tuple (e.g., expander walk). Specifically, they consider the code $\mathsf{Enc}(x) = \mathsf{Enc}_{\mathrm{walk}}(\mathsf{Enc}_{\mathrm{unique}}(x))$, where $\mathsf{Enc}_{\mathrm{unique}}\colon \mathbb{F}_p^n \to \mathbb{F}_p^{n'}$ is a unique-decodable code and $\mathsf{Enc}_{\mathrm{walk}}\colon \mathbb{F}_p^{n'} \to \mathbb{F}_p^N$ is an expander-walk direct sum code. Then, they show the list-decodability of Enc using the following argument: Run the decoder described above, which outputs a set L containing approximate codewords of $\mathsf{Enc}_{\mathrm{unique}}$. Thereafter, run the unique decoder of $\mathsf{Enc}_{\mathrm{unique}}$ on each $x \in L$, which yields a list of all codewords of Enc that is $(1/p-\varepsilon)$-close to the input string $\tilde{y}$.

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