A Study of Error Reduction Polynomials

Gil Cohen[*] Dean Doron[†] Tomer Manket[*]

gil@tauex.tau.ac.il deand@bgu.ac.il tomermanket@mail.tau.ac.il

Edward Pyne[‡] Yichuan Wang[§] Tal Yankovitz[*]

epyne@mit.edu yichuan-21@mails.tsinghua.edu.cn talyankovitz@mail.tau.ac.il

Abstract

Error reduction procedures play a crucial role in constructing weighted PRGs [PV21, CDR+21], which are central to many recent advances in space-bounded derandomization. The fundamental method driving error reduction procedures is the Richardson iteration, which is adapted from the literature on fast Laplacian solvers. In the context of space-bounded derandomization, where time is not the primary concern, can additional insights from optimization theory lead to improved error reduction procedures?

The results of this paper reveal an inherent barrier for using optimization-based techniques for error reduction in the context of space-bounded derandomization. To this end, we develop an abstract framework for error reduction based on polynomials which in particular encompasses all optimization-based error reduction techniques, and consequently, puts a limit on constructing improved weighted PRGs based on current approaches. Our work also sheds light on the necessity of negative weights within existing methods.

From the technical viewpoint, we establish lower bounds on various important parameters of error reduction polynomials. This includes a lower bound on the degree d of the polynomial, and an $n^{\Omega(d)}$ lower bound on L_1-norm of an n-variate polynomial. These lower bounds hold both when there are no "correlations" between different approximations and in the presence of such. A delicate use of these correlations has recently been exploited for constructing improved weighted PRGs for various restricted models [CHL+23].

[*] Tel Aviv University. Supported by ERC starting grant 949499.

[†] Ben-Gurion University of the Negev.

[‡] MIT. Supported by a Jane Street Graduate Research Fellowship.

[§] Institute for Interdisciplinary Information Sciences, Tsinghua University.

Contents

1 Introduction

BPL vs. **L**—a fundamental problem in computational complexity theory [AKL+79, BCP83, Jun81]—centers on whether randomness enhances the computational capabilities of space-bounded algorithms. Specifically, the question is whether every probabilistic algorithm can be fully derandomized with only a constant factor increase in space. Despite numerous attempts to resolve this problem, it remains open, although it is generally believed that **BPL** = **L**, a conclusion supported by plausible circuit lower bounds [KvM02, DT23, DPT24]. Moreover, unlike the time setting, there are currently no known barriers to the unconditional derandomization of **BPL**. The primary strategy for tackling **BPL** vs. **L** involves the use of pseudorandom generators (PRGs), specifically for the corresponding nonuniform model of read-once branching programs, abbreviated as ROBPs[1]. In his seminal paper, Nisan [Nis92] constructed an explicit ε-error PRG for ROBPs of length n and width w, with a seed length of $O(\log(n) \cdot \log(\frac{nw}{\varepsilon}))$. Most of the PRGs designed for general ROBPs draw inspiration from Nisan's prototype. We refer the reader to Hoza's survey [Hoz22] on recent advances in derandomizing space-bounded computation.

A common feature of these generators is their recursive design, where a PRG for length-n ROBPs is constructed using PRGs for length-$\frac{n}{2}$ ROBPs. This recursive construction leads to "error deterioration"; that is, if the error for the half-length PRG is ε, the error of the resulting PRG, for length n, at least doubles. Consequently, it is necessary to start with an initial error of roughly $\frac{\varepsilon}{n}$ to achieve a final error of ε. Specifically, an error of $\frac{1}{n}$ must be maintained throughout the recursion, even if the target is merely a constant error ε, which is indeed the main setting of parameters for derandomization. This error deterioration underpins the $\log^2 n$ dependency of the seed length in Nisan's PRG and its subsequent developments.

1.1 The BCG Construction

Motivated by the challenge of constructing a PRG for length-n ROBPs with a logarithmic dependence on n, Braverman, Cohen, and Garg [BCG18] directed their focus toward the *error parameter*. Their objective was to design a PRG that provides a slower error deterioration. Although they successfully improved the control on the error parameter, achieving near-optimal error dependence on ε in the seed length, the quadratic dependence $\log^2 n$ persisted in their seed length for a different, more subtle reason.

Interestingly, the BCG construction is not a PRG per se but rather a variant they dubbed a *weighted* PRG (WPRG). In a weighted PRG, each seed is associated with a weight, which

[1] We use the standard model of (standard-order) ROBPs as can be found, e.g., in [Hoz22, Definition 3.2.1].

is a real number that can be either positive or negative and is unbounded in absolute value. The desired ε-approximation is achieved through a weighted sum, rather than an average, as in traditional PRGs. More precisely, a weighted PRG is a function $\mathsf{WPRG}\colon \{0,1\}^s \to \{0,1\}^n \times \mathbb{R}$ with the following property: For any length-n ROBP P, the sum of the real-valued weights—corresponding to the seeds that lead to an accepting state of P—provides an ε-approximation of the probability that n truly uniform bits will lead to an accepting state (see [Hoz22, Defintion 6]).

The BCG construction is also influenced by Nisan's PRG but is significantly more complex. This complexity makes it challenging to integrate with other work, particularly when seeking improvements for constrained models such as regular ROBPs. Moreover, the role of the somewhat enigmatic negative weights remains elusive and unclear. Chattopadhyay and Liao [CL20] managed to simplify the construction slightly, yet it still follows to the general ideas proposed by BCG and remains highly intricate (see also [Koz20]).[2]

1.2 Error Reduction via Richardson Iteration

In concurrent and independent works, Pyne and Vadhan [PV21] and Cohen, Doron, Renard, Sberlo, and Ta-Shma [CDR$^+$21] devised an alternative method to that of BCG for achieving improved error dependence. Rather than constructing a WPRG from scratch, these papers have introduced (the same) error reduction procedure – a technique for transforming a PRG for length-n, width-w ROBPs with a seed length of $s = s(n,w)$ into an ε-error WPRG with a seed length of $\widetilde{O}(s+\log\frac{1}{\varepsilon})$. Crucially, this method requires the initial PRG to have an error of approximately $\frac{1}{n}$. When instantiated with Nisan's PRG, this approach produces results that are quantitatively comparable to—and slightly improve upon—those of BCG. Consequently, Hoza improved these constructions by eliminating all doubly-logarithmic factors [Hoz21].

The error-reduction procedures that were devised in these works draws on the Richardson iteration method, a strategy adopted from nearly linear-time Laplacian solvers (starting from [ST04]), which is also commonly employed in numerical linear algebra and optimization. This method marks a departure from the recursive powering approach central to Nisan's PRG and its variants, including BCG.

At its core, the Richardson iteration is an iterative process. The error reduction approach, first used in the context of space-bounded computation (in the non black-box setting) by Murtagh, Reingold, Sidford, and Vadhan [MRSV17] begins by deriving a real-valued polynomial in non-commutative variables from this iterative method, which we refer to as the

[2] It is worth mentioning that optimal *hitting set generators* in the low-error regime (which are a weaker variant of PRGs and WPRGs), turned out to have simple constructions which utilize different properties and techniques [HZ20].

Richardson polynomial. More specifically, a family of polynomials can be generated based on the number k of iterations performed. The polynomial from the k-th iteration, which has a degree of $O(k)$, is denoted as Rich_k. For the sake of simplicity in this informal discussion, we assume all layers of the branching program are the same, represented by a common transition matrix A. Our objective is then to approximate the matrix A^n. Under this simplification, Rich_k is a polynomial in n non-commutative variables and meets the following condition: If $\widetilde{A^2}, \dots, \widetilde{A^n}$ are ε-approximations for the respective powers of A, defined such that $\|\widetilde{A^j} - A^j\| \le \varepsilon$ for all j (denoted $\widetilde{A^j} \approx_\varepsilon A^j$), then

$$\mathsf{Rich}_k\left(A, \widetilde{A^2}, \dots, \widetilde{A^n}\right) \approx_{\varepsilon_k} A^n,$$

where $\varepsilon_k \approx (n \cdot \varepsilon)^k$. The choice of norm is arbitrary provided that it is sub-multiplicative.

The second phase of the error-reduction procedure involves deriving the final WPRG from the Richardson polynomial Rich_k and the PRGs used to obtain the approximations $\widetilde{A^2}, \dots, \widetilde{A^n}$. This step incorporates a *second* PRG for ROBPs aimed at reusing the seeds of the original PRG. We shall not elaborate further on this step here (we do so in Section 2.3) but highlight that this method can be applied with any polynomial, not exclusively Richardson polynomials. Properties of the polynomial, such as its degree, sparsity, and the L_1-norm of the coefficient vector, can be used for establishing bounds on the final error of the WPRGs.

Error reduction procedures: A broader context

Error reduction procedures are not limited to space-bounded derandomization and are prevalent throughout theoretical computer science. For instance, Raz, Reingold, and Vadhan devised an error-reduction procedure for seeded extractors [RRV99]. More recently, Ta-Shma's breakthrough work on constructing small-bias sets relied on an innovative method to reduce the error, or bias, of an existing small-bias set [Ta-17]. Furthermore, techniques such as derandomized squaring [RV05, MRSV17, MRSV19, AKM$^+$20, HPV21, CCMP23], distance amplification in error-correcting codes [AL96, KMRZS17, CY21, CY22], and gap amplification in PCPs [Din07] are all examples of error-reduction procedures. These procedures are evaluated based on the spectral gap, code distance, and PCP gap, respectively. Notably, even Reingold's Theorem, $\mathbf{SL} = \mathbf{L}$, [Rei08] involves a bespoke error reduction procedure tailored to the specific graph in question.

In the realm of space-bounded derandomization, the focus on the error parameter and specifically error-reduction procedures forms the foundation of many recent advancements in this area (e.g., [BCG18, HPV21, PV21, Hoz21, CHL$^+$23]). The incorporation of ideas from fast Laplacian solvers and general optimization theory has injected new and exciting results into the field. Consequently, the primary aim of this paper is to explore whether further results can be expected from these disciplines or if we have encountered a fundamental limit. Unfortunately, our work suggests that the latter scenario is more likely.

A closer look at Richardson iteration

So far we have not discussed the way in which the Richardson polynomials, Rich_k, are generated. Indeed, these are obtained in a somewhat roundabout way. The Richardson iteration is actually a way of improving the approximation of a matrix *inverse*. More precisely, given an invertible matrix L and an initial approximation $\widetilde{L^{-1}}$ to its inverse, the Richardson iteration refines it into a more accurate approximation of L^{-1}, where with each iteration a stronger approximation is guaranteed.

To utilize this in matrix *powering* (the more general case of iterated matrix multiplication is similar), one constructs an $(n+1) \times (n+1)$ block matrix L, with I_w on the diagonal and $-A$ on the sub-diagonal. Clearly, L is an invertible matrix where, crucially, the bottom-left block of L^{-1} equals A^n. Moreover, L^{-1} itself is lower triangular, with the (i,j)-th block being A^{i-j}. With this in mind, the approximation $\widetilde{L^{-1}}$ is then naturally constructed by setting each (i,j)-th block, lying below the diagonal, to $\widetilde{A^{i-j}}$. Applying Richardson iteration to L and $\widetilde{L^{-1}}$ yields a refined approximation of L^{-1}, from which we extract the bottom-left block as the improved approximation for A^n.

For example, one iteration of the Richardson iteration applied to L and $\widetilde{L^{-1}}$ takes the form $R_1 = 2\widetilde{L^{-1}} - \widetilde{L^{-1}}L\widetilde{L^{-1}}$. So, for approximating, say, the fourth power, A^4, one first constructs the matrices

$$L = \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ -A & I & 0 & 0 & 0 \\ 0 & -A & I & 0 & 0 \\ 0 & 0 & -A & I & 0 \\ 0 & 0 & 0 & -A & I \end{pmatrix} \qquad \widetilde{L^{-1}} = \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ A & I & 0 & 0 & 0 \\ \widetilde{A^2} & A & I & 0 & 0 \\ \widetilde{A^3} & \widetilde{A^2} & A & I & 0 \\ \widetilde{A^4} & \widetilde{A^3} & \widetilde{A^2} & A & I \end{pmatrix}. \tag{1}$$

It then follows by a direct calculation that the bottom-left block of R_1 is

$$A\widetilde{A^2}A + A^2\widetilde{A^2} - (\widetilde{A^2})^2.$$

Put differently,

$$\mathsf{Rich}_1(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) = \boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_1 + \boldsymbol{x}_1^2\boldsymbol{x}_2 - \boldsymbol{x}_2^2. \tag{2}$$

Using a similar calculation, the values of Rich_1 for any given n can be determined:

$$\mathsf{Rich}_1(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \boldsymbol{x}_{n-1}\boldsymbol{x}_1 + \sum_{i=1}^{n-2} \boldsymbol{x}_i\,(\boldsymbol{x}_{n-1-i}\boldsymbol{x}_1 - \boldsymbol{x}_{n-i}). \tag{3}$$

For $k \geq 1$, the Richardson iteration induces a very simple polynomial, namely, the polynomial corresponding to k iterations is given by

$$R_k = \sum_{i=0}^{k} \left(I - \widetilde{L^{-1}}L\right)^i \widetilde{L^{-1}}. \tag{4}$$

From this, with some effort, one can construct the Richardson polynomial, Rich_k, for any $k \geq 2$. For instance, the bottom-left block of R_2, set with $n = 5$, is given by

$$A\left(\widetilde{A^3}A - \widetilde{A^4} + \left(A^2 - \widetilde{A^2}\right)^2\right) + \widetilde{A^2}\left(\widetilde{A^2}A - \widetilde{A^3}\right) + \widetilde{A^3}\left(A^2 - \widetilde{A^2}\right) +$$
$$\widetilde{A^4}A + 2\left(A^2 - \widetilde{A^2}\right)\left(\widetilde{A^2}A - \widetilde{A^3}\right).$$

Thus,

$$\mathsf{Rich}_2(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_5) = \boldsymbol{x}_1\left(\boldsymbol{x}_3\boldsymbol{x}_1 - \boldsymbol{x}_4 + \left(\boldsymbol{x}_1^2 - \boldsymbol{x}_2\right)^2\right) + \boldsymbol{x}_2\left(\boldsymbol{x}_2\boldsymbol{x}_1 - \boldsymbol{x}_3\right) + \boldsymbol{x}_3\left(\boldsymbol{x}_1^2 - \boldsymbol{x}_2\right) +$$
$$\boldsymbol{x}_4\boldsymbol{x}_1 + 2\left(\boldsymbol{x}_1^2 - \boldsymbol{x}_2\right)\left(\boldsymbol{x}_2\boldsymbol{x}_1 - \boldsymbol{x}_3\right).$$

Richardson iteration via gradient descent

Equation (4) can be derived from an optimization point of view. To see this, assume as before that we are given a crude approximation $\widetilde{L^{-1}}$ to L^{-1} as well as L itself, and we wish to find a better approximation for L^{-1}, making use of the fact that the crude approximation $\widetilde{L^{-1}}$ is available to us. Assume for simplicity that L is positive-definite (otherwise, consider $L^{\mathsf{T}}L$ instead of L in what follows). Finding a better approximation for L^{-1} effectively means that we would like to approximate, given a vector b, the solution $x^\star$ to the system of linear equations $Lx^\star = b$, in a way which is, in a sense, independent of b.

We can now utilize the fact that we have access to $\widetilde{L^{-1}}$ and reduce the condition number of the linear system, by instead solving $\widetilde{L^{-1}}Lx^\star = \widetilde{L^{-1}}b \triangleq b_0$. The iterative approach now presents itself: Starting with the initial point $x_0 = \alpha\widetilde{L^{-1}}b$, set according to some parameter $\alpha > 0$ which will also serve as our step size, we generate a sequence $x_0, x_1, \ldots, x_k$ of solutions, where we wish to minimize the error vectors $x_i - x^\star$. It makes sense to use the geometry of the problem and measure this error vector in a suitable norm, in particular, $e_i = \|x_i - x^\star\|_{\widetilde{L^{-1}}L}$. Thus, we have the optimization problem in which we wish to minimize $f(x) = \frac{1}{2}x^{\mathsf{T}}\widetilde{L^{-1}}Lx - b_0^{\mathsf{T}}x$. This can be done using gradient descent, noting that the gradient $\nabla f(x) = \widetilde{L^{-1}}Lx - b_0$ is convex. The unique minimum is thus obtained when $\nabla f(x) = 0$, namely, at $x^\star$.

The gradient descent iteration $x_{i+1} = x_i - \alpha\nabla f(x_i)$ then becomes $x_{i+1} = (I - \alpha\widetilde{L^{-1}}L)x_i +$

αb_0, and so

$$x_k = \alpha \sum_{i=0}^{k} \left(I - \alpha \widetilde{L^{-1}} L\right)^i \widetilde{L^{-1}} b \triangleq R_k(\alpha) b. \tag{5}$$

Importantly, $R_k(\alpha)$ is indeed independent of the choice of b. Note that $x_k - x^\star = P_k(x_0 - x^\star)$, where $P_k = (I - \alpha \widetilde{L^{-1}} L)^k$. Thus, the error e_k is dominated by $\|P_k\| \le \|I - \alpha \widetilde{L^{-1}} L\|^k$. The familiar Richardson iteration used in the context of space-bounded derandomization amounts to taking $\alpha = 1$, but optimizing α based on the system's condition number leads to better convergence rate, although not in a way that would improve upon the space-bounded literature.

In fact, it is known that *any* first-order method, in which

$$x_{i+1} \in x_0 + \operatorname{Span}\left\{\nabla f(x_0), \ldots, \nabla f(x_i)\right\},$$

can be expressed as a *polynomial iteration* akin to Equation (5) and in which $x_k - x^\star = \overline{P}_k(x_0 - x^\star)$ for some degree-k polynomial $\overline{P}_k$ (see [dST$^+$21, Section 2]). This observation allows one to study optimal polynomials, and indeed, using *shifted Chyebyshev polynomials* gives rise to better iterations, improving quadartically on the degree of R_k but again, not in a way that would improve upon the state-of-the-art in space-bounded derandomization.

2 Error Reduction: The Abstract Framework

The above discussion raises a natural question: Can we gain by moving beyond first-order methods? Richardson iteration and gradient descent are typical first-order methods suitable for nearly-linear time algorithms. We, however, are focused on minimizing *space*, which allows us the flexibility to employ second- or higher-order methods. Specifically, it is conceivable to design a convex program that is more sophisticated than the quadratic program previously discussed, embedding more detailed information about L and $\widetilde{L^{-1}}$ into its geometry. Subsequently, an approximate solution could be derived using a second- or higher-order method whose steps make a better use of L and $\widetilde{L^{-1}}$. This approach would likely result in a new polynomial, similar to what was mentioned earlier.

Informally speaking, the main result of this work identifies a barrier to developing error-reduction procedures based on this approach. Our work suggests that the Richardson iteration is essentially *optimal* among error-reduction procedures derived from optimization-based approaches. To formalize, and prove, this somewhat vague assertion, we will develop an abstract framework for error reduction in the next section. This framework has the structure of an affine subspace, with each point representing a distinct error-reduction procedure, one of which is the familiar Richardson iteration based error reduction. This abstract approach will enable us to consider, and argue about, all error-reduction procedures simultaneously.

2.1 The Polynomial Error-Reduction Framework

Our first key observation is that iterative methods invariably generate polynomials, with the polynomial's degree corresponding to the number of iterations performed. The abstract framework that we introduce encompasses *all* bounded-degree polynomials, not just those derived from optimization theory. To describe our framework, and gain some intuition, we will begin with a sequence of examples. Our examples, and the framework in general, focuses on error reduction for *numbers* rather than for *matrices*. Our work reveals that the barrier for achieving better-than-Richardson error-reduction procedures is not due to the non-commutative nature of the matrices involved. Instead, it is a numerical phenomenon that manifests even in the case of numbers.[3]

Degree-3 error-reduction polynomials for approximating a 4-th power

Let a represent the number—rather than a matrix—whose n-th power we aim to approximate. As an initial example, consider the simple case where $n = 4$. In this scenario, we are given a along with approximations for its powers, which we denote as $\widetilde{a^2}$, $\widetilde{a^3}$, and $\widetilde{a^4}$. Our objective is to devise a polynomial that, when provided with these four values as inputs, produces a better approximation than those initially given. Evidently, a polynomial of degree 4—or indeed a polynomial of degree n in general—can simply compute a^n while disregarding the provided approximations. However, this approach results in an error-reduction procedure that is inefficient due to the significant computational cost associated with higher degrees (see Section 2.3). Specifically, such error reduction may completely disregard the given approximation, rendering it moot. Therefore, our goal is to develop a polynomial of lower degree. In this example, where n is small to begin with, we aim to construct a polynomial of degree $d = 3$.

How can we find such a polynomial? As a starting point, we consider *all* potential polynomials, specifically, any general degree-3 polynomial $\mathsf{Q}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) \in \mathbb{R}[\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4]$, or complex-valued polynomials far that matter. However, upon reflection, we should focus only on those monomials that, after substituting a for $\boldsymbol{x}_1$ and $\widetilde{a^j}$ for $\boldsymbol{x}_j$ (where $j = 2, 3, 4$), result in an "effective degree" of a being 4 (see Lemma 3.5). Therefore, the polynomial can be expressed as a combination of basis monomials $\boldsymbol{x}_1^2\boldsymbol{x}_2$, $\boldsymbol{x}_1\boldsymbol{x}_3$, $\boldsymbol{x}_2^2$, and $\boldsymbol{x}_4$ (whereas monomials such as $\boldsymbol{x}_1\boldsymbol{x}_2$, $\boldsymbol{x}_1\boldsymbol{x}_3$ and $\boldsymbol{x}_1^3$ are ignored), taking the form

$$\mathsf{Q}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) = A\boldsymbol{x}_1^2\boldsymbol{x}_2 + B\boldsymbol{x}_1\boldsymbol{x}_3 + C\boldsymbol{x}_2^2 + D\boldsymbol{x}_4$$

[3] One can be convinced that reducing errors when dealing with numbers in the range $[0, 1]$ is indeed simpler than the corresponding task for general stochastic matrices. The latter are equivalent to general ROBPs, but may involve a larger alphabet, Σ. However, well-known reductions can be applied to convert these to a binary alphabet.

for some coefficients A, B, C, and D. With this formulation, we seek for the constraints, under which, if $|\widetilde{a^j} - a^j| \leq \varepsilon_0$ for $j = 2, 3, 4$, then

$$\mathsf{Q}(a, \widetilde{a^2}, \widetilde{a^3}, \widetilde{a^4}) \ll \varepsilon_0. \tag{6}$$

Let us denote $\widetilde{a^j} = a^j + \boldsymbol{\varepsilon}_j$. Then,

$$\begin{aligned}\mathsf{Q}(a, \widetilde{a^2}, \widetilde{a^3}, \widetilde{a^4}) &= \mathsf{Q}(a, a^2 + \boldsymbol{\varepsilon}_2, a^3 + \boldsymbol{\varepsilon}_3, a^4 + \boldsymbol{\varepsilon}_4) \\ &= Aa^2(a^2 + \boldsymbol{\varepsilon}_2) + Ba(a^3 + \boldsymbol{\varepsilon}_3) + C(a^2 + \boldsymbol{\varepsilon}_2)^2 + D(a^4 + \boldsymbol{\varepsilon}_4).\end{aligned}$$

We have now expressed Q in terms of $\boldsymbol{\varepsilon}_2, \boldsymbol{\varepsilon}_3, \boldsymbol{\varepsilon}_4$, and a, rather than $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4$, which enables us to better concentrate on the *errors* rather than on the approximations. Expanding Q with respect to these variables yields

$$\mathsf{Q}(a, \widetilde{a^2}, \widetilde{a^3}, \widetilde{a^4}) = (A + B + C + D)a^4 + (A + 2C)a^2\boldsymbol{\varepsilon}_2 + Ba\boldsymbol{\varepsilon}_3 + C\boldsymbol{\varepsilon}_2^2 + D\boldsymbol{\varepsilon}_4. \tag{7}$$

Since our goal is to approximate a^4, the affine equation

$$A + B + C + D = 1 \tag{8}$$

must be satisfied [4].

To formalize Equation (6), in this instance, we require the coefficients of the *linear* error terms, $\boldsymbol{\varepsilon}_2, \boldsymbol{\varepsilon}_3$ and $\boldsymbol{\varepsilon}_4$, to vanish, leaving us only with the quadratic error term. The rationale is that in developing an error-reduction procedure, we cannot assume any relationship between the errors $\boldsymbol{\varepsilon}_2, \boldsymbol{\varepsilon}_3$, and $\boldsymbol{\varepsilon}_4$. Based on this observation, we establish the principle that underlies our methodology, treating these errors as algebraically independent, or formal, variables. We will reexamine this foundational assumption in our abstract formal framework in Section 2.4.

At any rate, this approach introduces three additional *homogeneous* linear constraints to the affine equation (8): $A + 2C = 0$ and $B = D = 0$. Crucially, these linear equations are *independent* of a. This is the way in which the remarkable existence of black-box error reduction—unaffected by the specific number provided (a principle confirmed by Richardson iteration)—manifests itself in our abstract framework. Indeed, if the equations were dependent on a, it would compromise the black-box nature of the error reduction.

Returning back to the specific example, these four conditions completely determine the polynomial: $A = 2$, $B = D = 0$, and $C = -1$, resulting in

$$\mathsf{Q}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) = 2\boldsymbol{x}_1^2\boldsymbol{x}_2 - \boldsymbol{x}_2^2. \tag{9}$$

[4] Technically, a RHS value close to 1, within the required approximation guarantee, would also be acceptable as it would lead to the desired approximation. However, our results are applicable to any nonzero value; therefore, we naturally choose to set the RHS to 1.

Looking back at Equation (7), and using that $C = -1$ and that $|\boldsymbol{\varepsilon}_2| \leq \varepsilon_0$, the error produced by this polynomial is $|\mathsf{Q}(a, \widetilde{a^2}, \widetilde{a^3}, \widetilde{a^4}) - a^4| = \boldsymbol{\varepsilon}_2^2 \leq \varepsilon_0^2$, and so we have reduced the error quadratically. Interestingly, due to its uniqueness, this is the exact polynomial obtained through the Richardson iteration as described in Equation (2), adjusted for commutativity. Here, however, we have constructed this polynomial not through optimization techniques but using elementary linear algebra principles. This approach offers a significant advantage: we now understand that this is the *only* degree-3 polynomial capable of reducing the error quadratically.

The uniqueness of this particular polynomial implies that there is no degree-2 error-reduction polynomial where all linear error terms vanish. With the ideas discussed above, the diligent reader might want to prove that no such polynomial exists for any $n > 2$. Before drawing further general conclusions, let us consider a second example.

Degree-3 error-reduction polynomials for approximating a 5-th power

We can do the same process for error reduction of the 5-th power of a number. This time, the polynomial takes the general form

$$\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_5) = A\boldsymbol{x}_1^2\boldsymbol{x}_3 + B\boldsymbol{x}_1\boldsymbol{x}_4 + C\boldsymbol{x}_1\boldsymbol{x}_2^2 + D\boldsymbol{x}_2\boldsymbol{x}_3 + E\boldsymbol{x}_5.$$

Using the same notation as in Section 2.1, we can express $\mathsf{Q}(a, \widetilde{a^2}, \ldots, \widetilde{a^5})$ in terms of the variables $\boldsymbol{\varepsilon}_2, \ldots, \boldsymbol{\varepsilon}_5$, and a to get

$$(A + B + C + D + E)a^5 + (2C + D)a^3\boldsymbol{\varepsilon}_2 + (A + D)a^2\boldsymbol{\varepsilon}_3 + Ba\boldsymbol{\varepsilon}_4 + E\boldsymbol{\varepsilon}_5 + Ca\boldsymbol{\varepsilon}_2^2 + D\boldsymbol{\varepsilon}_2\boldsymbol{\varepsilon}_3.$$

As in Equation (8), we encounter the affine constraint $A+B+C+D+E = 1$. If our aim here is also to eliminate the linear error terms, we obtain the following linear homogeneous constraints: $2C + D = 0$, $A + D = 0$ as well as $B = E = 0$. These constraints yield a unique solution in which $A = 2$, $C = 1$, and $D = -2$, resulting in the polynomial

$$\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_5) = 2\boldsymbol{x}_1^2\boldsymbol{x}_3 + \boldsymbol{x}_1\boldsymbol{x}_2^2 - 2\boldsymbol{x}_2\boldsymbol{x}_3. \tag{10}$$

Given its uniqueness, this polynomial must be the one derived from the Richardson iteration, for one iteration [5] set with $n = 5$. The error associated with this polynomial is given by

$$\left|\mathsf{Q}(a, \widetilde{a^2}, \ldots, \widetilde{a^5}) - a^5\right| = \left|a\boldsymbol{\varepsilon}_2^2 - 2\boldsymbol{\varepsilon}_2\boldsymbol{\varepsilon}_3\right| \leq 3\varepsilon_0^2.$$

[5] By examining Equations (1) and (4) we see that the k-th Richardon polynomial, Rich_k, has degree $2k+1$ in its input matrices $A, \widetilde{A^2}, \ldots, \widetilde{A^n}$ and by extension in $a, \widetilde{a^2}, \ldots, \widetilde{a^n}$ when considering numbers.

Degree-3 error-reduction polynomials for approximating a 6-th power

The situation grows more intriguing when we consider the case $n = 6$. In this scenario, a general degree-3 polynomial takes the form

$$\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_6) = A\boldsymbol{x}_1^2\boldsymbol{x}_4 + B\boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_3 + C\boldsymbol{x}_2^3 + D\boldsymbol{x}_2\boldsymbol{x}_4 + E\boldsymbol{x}_3^2. \tag{11}$$

An observant reader might notice that we have overlooked the perfectly valid monomials $\boldsymbol{x}_1\boldsymbol{x}_5$ and $\boldsymbol{x}_6$. This omission is due to $\boldsymbol{x}_5$ and $\boldsymbol{x}_6$ only appearing in a single monomial; thus, anticipating future developments, the coefficients of these terms must be set to zero. By a computation similar to the previous ones, we derive the linear system:

$$\begin{aligned} &A + B + C + D + E = 1,\\ &B + 3C + D = 0,\\ &B + 2E = 0,\\ &A + D = 0. \end{aligned} \tag{12}$$

This time, however, there is no unique solution. Instead, we find a one-dimensional affine subspace of solutions, where the family of error-reduction polynomials is parameterized by $A \in \mathbb{R}$ and defined by Equation (11) along with: $B = 6 - 2A$, $C = A - 2$, $D = -A$, and $E = A - 3$. For example, setting $A = 3$ results in a fairly sparse polynomial as $B = E = 0$, yielding the polynomial

$$\mathsf{Q}_3(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_6) = 3\boldsymbol{x}_1^2\boldsymbol{x}_4 + \boldsymbol{x}_2^3 - 3\boldsymbol{x}_2\boldsymbol{x}_4.$$

The error resulting from this choice is

$$\left|\mathsf{Q}_3(a, \widetilde{a^2}, \widetilde{a^4}) - a^6\right| = \left|3a^2\boldsymbol{\varepsilon}_2^2 + \boldsymbol{\varepsilon}_2^3 - 3\boldsymbol{\varepsilon}_2\boldsymbol{\varepsilon}_4\right| \leq 6\varepsilon_0^2 + O(\varepsilon_0^3). \tag{13}$$

Note that we cannot expect to improve on the triangle inequality used in the last inequality, given that we have no control over the relations between the errors $\boldsymbol{\varepsilon}_2$, $\boldsymbol{\varepsilon}_4$, and the value a. Given this observation, could there be a more optimal point within this affine subspace, namely, a better degree-3 error-reduction polynomial? To answer this, we evaluate the error term for general A, to find that

$$\left|\mathsf{Q}_A(a, \widetilde{a^2}, \widetilde{a^4}) - a^6\right| = \left|(6 - 2A)a\boldsymbol{\varepsilon}_2\boldsymbol{\varepsilon}_3 - A\boldsymbol{\varepsilon}_2\boldsymbol{\varepsilon}_4 + 3(A - 2)a^2\boldsymbol{\varepsilon}_2^2 + (A - 3)\boldsymbol{\varepsilon}_3^2\right| + O(\varepsilon_0^3).$$

Based on the above reasoning, our aim should be to minimize the expression:

$$|6 - 2A| + |A| + |3(A - 2)| + |A - 3| = 3|A - 3| + 3|A - 2| + |A|.$$

The minimum of this function is attained at $A = 2$, yielding a bound of $5\varepsilon_0^2 + O(\varepsilon_0^3)$, which improves upon the $6\varepsilon_0^2 + O(\varepsilon_0^3)$ bound found in Equation (13) when optimizing for sparsity. Interestingly, by inspecting Equation (3), we see that this choice, $A = 2$, corresponds exactly to the Richardson polynomial.

We make one final comment before discussing the general case. It is clear that any error reduction procedure must receive a as input. Indeed, without it, and only relying on approximations, possibly including an approximation to a, denoted $\widetilde{a}$, there simply isn't enough information about a to compute, or better approximate, its n-th power. It is worth noting how this manifests itself in our abstract framework. In the particular example above, by referring to Equation (11) and substituting $\boldsymbol{x}_1$ with $a + \boldsymbol{\varepsilon}_1$, we introduce an additional homogeneous constraint to the system given in Equation (12), corresponding to $\boldsymbol{\varepsilon}_1$, which is $2A + B = 0$. However, recall that the original system requires $B = 6 - 2A$ which contradicts the new constraint. This contradiction can be shown to be a general phenomenon—the presence of an error $\boldsymbol{\varepsilon}_1$ in the approximation for a invariably conflicts with the affine constraint.

The general case

The affine space of error-reduction polynomials of degree d in n variables, where all error terms of degree at most t vanish, is denoted by $\mathcal{E}(n, d, t)$. The above discussion indicates that $\mathcal{E}(n, 2, 1) = \emptyset$ for any $n > 2$, and that $\mathcal{E}(5, 3, 1)$ consists solely of the Richardson polynomial, thus $\dim \mathcal{E}(5, 3, 1) = 0$. Moreover, $\dim \mathcal{E}(6, 3, 1) = 1$ and similarly, $\dim \mathcal{E}(7, 3, 1) = 1$ whereas $\dim \mathcal{E}(8, 3, 1) = 2$.

As we move to higher degrees, new phenomena emerge. For instance, $\dim \mathcal{E}(5, 4, 1) = 1$, and the polynomial parameterized by A is given by

$$\mathsf{Q}_A(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_5) = A\boldsymbol{x}_1^3\boldsymbol{x}_2 + (2 - A)\boldsymbol{x}_1^2\boldsymbol{x}_3 + (1 - A)\boldsymbol{x}_1\boldsymbol{x}_2^2 + (A - 2)\boldsymbol{x}_2\boldsymbol{x}_3.$$

Interestingly, aiming at minimizing the error as in the 6-th power case, here an *interval* for A exists where the error is optimized, specifically, $A \in [1, 2]$. We also observe that the choice of $A = 0$, which results in the unique degree-3 polynomial (compare with Equation (10)), is sub-optimal among degree-4 polynomials. Specifically, the optimal polynomials in $\mathcal{E}(5, 4, 1)$ *do not correspond to a Richardson polynomial*, which is always of odd degree.

In general, $\mathcal{E}(n, d, t)$ is spanned by monomials of the form $\boldsymbol{x}_1^{e_1} \cdots \boldsymbol{x}_n^{e_n}$, where $e_1 + \cdots + e_n \leq d$, subject to certain linear constraints. More specifically, every polynomial in $\mathcal{E}(n, d, t)$ must satisfy one affine constraint (as in Equation (8)) and a total of approximately n^t homogeneous constraints—one for each nonempty monomial in $\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_n$ up to degree t. Therefore, even taking into account the fact that the effective degree of the monomials should equal n, it is plausible that as long as $d \gg t$, the space $\mathcal{E}(n, d, t)$ will contain numerous error-reduction polynomials. Indeed, this is confirmed simply by comparing the degrees of freedom with the number of homogeneous constraints, provided that these constraints do not contradict the affine constraint.

2.2 Our Results

This wealth of error-reduction polynomials suggests that there may be other polynomials which outperform the corresponding Richardson polynomial within the same space, $\mathcal{E}(n, d, t)$. For the purpose of error reduction, the properties we consider important are the polynomial's degree, sparsity, and the L_1-norm of its coefficients. We will elaborate on the reasons for this in the next section (Section 2.3). However, even at this stage, we can present our main results, effectively demonstrating that the Richardson polynomial is essentially optimal for space-bounded derandomization. More precisely, we prove that indeed eliminating the error terms of degree at most t requires a degree $d > t$, and provide lower bounds on the sparsity and L_1-norm of *all* error-reduction polynomials. Interestingly, our sparsity lower bound is proven only for the non-commutative setting.

Theorem 2.1 (main result; informal)**.** *For every $n > d > t \geq 1$ and $\mathsf{Q} \in \mathcal{E}(n, d, t)$ it holds that*

$$L_0(\mathsf{Q}) \geq \frac{\binom{n}{t}}{\binom{d}{t}} \geq \left(\frac{n}{d}\right)^t,$$

$$L_1(\mathsf{Q}) \geq 2\left(\frac{n}{d}\right)^t - 1,$$

where the lower bound on $L_0(\mathsf{Q})$ holds in the non-commutative setting. Moreover, for every $t \geq d$, $\mathcal{E}(n, d, t) = \emptyset$.

The proofs for the bounds summarized in Theorem 2.1 are detailed in Proposition 4.1, Theorem 5.5, and Theorem 6.3, which address the degree bound, L_1-bound, and L_0-bound, respectively.

For applications to space-bounded derandomization, one considers n as the primary parameter, tending towards infinity, while t increases more gradually with n, in particular, $t = \mathrm{poly}(\log n)$ (see Section 2.3). In this context, Theorem 2.1 suggests that the error-reduction polynomial should have a degree $d > t$, with its sparsity and L_1-norm being at least $n^{\Omega(t)}$. Moreover, achieving significantly reduced norms is only possible with very high degrees, specifically $d = n^{1-o(1)}$, which are infeasible in the space-bounded setting.

To conclude, we observe another corollary from our abstract framework: error reduction procedures that rely on polynomials—specifically, those derived from optimization theory—yield *weighted* PRGs rather than standard PRGs. The presence of negative numbers directly stems from the homogeneous linear constraints that must be met. Although intuitively clear, we believe this formalizes and clarifies the essential role of negative weights in error-reduction procedures, independent of their method of construction.

Correlated errors. As noted, Section 2.3 presents how error-reduction polynomials are utilized in the error-reduction procedure. In Section 2.4, inspired by the recent work of Chen, Hoza, Lyu, Tal, and Wu [CHL$^+$23], we revisit our initial assumption regarding the independence of errors. This assumption proves to be excessively pessimistic since correlations between errors *can* indeed be introduced. Notably, such correlations already manifest in existing constructions as a natural consequence of the recursive structure of Nisan's PRG. This would require us to revise our abstract framework to include algebraic dependencies between the variables. Quantitative bounds comparable to those obtained in Theorem 2.1 continue to hold, although the proof is more challenging.

2.3 The WPRGs Error Reduction Framework

We are given an ε_0-PRG G_0 that fools ROBPs of length n, with seed length $s_0 = s_0(n, \varepsilon_0)$.[6] Say we are equipped with a suitable error-reduction polynomial

$$\mathsf{Q}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = \sum_{\mathbf{I} \in \mathcal{I}} c_{\mathbf{I}} \boldsymbol{x}^{\mathbf{I}},$$

and our guarantee is that if, for any $i \geq 2$ it holds that $\left\| \widetilde{A^i} - A_i \right\| \leq \varepsilon_0$, then

$$\left\| \mathsf{Q}(A, \widetilde{A^2}, \dots, \widetilde{A^n}) - A^n \right\| \leq \tau$$

for some guaranteed bound τ that depends on ε_0 and the parameters of Q discussed soon. We do not explicitly state which (sub-multiplicative) norm are we using, since different norms can be useful, and indeed various norms were used in the literature, depending on the model under consideration. When G_0 indeed fools arbitrary ROBPs, and we wish our error-reduced WPRG G to fool arbitrary ROBPs as well, then the matrices we should handle are arbitrary stochastic matrices. But of course, one can think of G_0 as fooling more *structured* ROBPs, say regular ROBPs. In this case, one needs to guarantee error reduction only for doubly stochastic matrices. One can observe that in order to establish *lower bounds* for general stochastic matrices, it suffices to handle real numbers.

The following parameters of Q will be paramount in establishing our results.

[6] Here and throughout, for simplicity, we suppress the dependence on the width w (or one can think of $w = O(1)$) and assume that the alphabet of G_0 is $\Sigma = \{0, 1\}$.

- The ℓ_1**-norm** of its coefficient vector, namely $L_1(\mathsf{Q}) = \sum_{\mathbf{I}\in\mathcal{I}} |c_{\mathbf{I}}|$.
- Its **total degree**, $d = \max_{\mathbf{I}\in\mathcal{I}} |\mathbf{I}|$, where for $\mathbf{I} = (i_1, \dots, i_n)$ we define $|\mathbf{I}| = \sum_{k=1}^{n} i_k$.
- Its **sparsity**, namely $L_0(\mathsf{Q}) = \#\{\mathbf{I} \in \mathcal{I} \colon c_{\mathbf{I}} \neq 0\}$.

We will use the parameter t to denote the number of **vanishing levels**. That is, the largest integer such that $\mathsf{Q}(a, \widetilde{a^2}, \dots, \widetilde{a^n})$ contains only monomials in $\boldsymbol{\varepsilon}_2, \dots, \boldsymbol{\varepsilon}_n$ with total degree at least $t+1$. We think of $t = t(d)$, and naturally, for some designated t (which would mean, for a designated ε), we would want to minimize d. As an example, Rich_k satisfies $d = 2k+1$, and $t = d$.

The error reduction step. Under any choice of sub-multiplicative norm, we have that

$$\left\|\mathsf{Q}(A, \widetilde{A^2}, \dots, \widetilde{A^n})\right\| \leq n \cdot L_1(\mathsf{Q}) \cdot \varepsilon_0^{t+1}.$$

However, the error reduction capabilities of Q on stochastic matrices or subclasses of such may be better than the bound above (possibly under a suitable choice of norm). We will consider a few examples from the space-bounded literature below.

The WPRG step. Equipped with G_0 and Q, we are ready to perform the error reduction, and get a WPRG G that fools the same function class, but with a better error dependence. The construction follows by evaluating each term $\widetilde{A^i}$ by a suitable instantiation of G_0. To approximate an entire *monomial*, rather than choosing independent seeds for each invocation of G_0, an "auxiliary PRG" G_{aux} (say, the [INW94] PRG) is employed. The auxiliary PRG needs to fool length-d ROBPs with alphabet $\{0,1\}^{s_0}$ and error $O\left(\frac{\varepsilon}{L_0(\mathsf{Q})\cdot L_1(\mathsf{Q})}\right)$. The seed for G then comprises a seed for G_{aux}, and an additional seed of length $O(\log L_0(\mathsf{Q}))$ to index a specific monomial.[7] Plugging-in the parameters for G_{aux},[8] the seed length of G then becomes

$$s(n, \varepsilon_0, \varepsilon) = O\left(s_0(n, \varepsilon_0) + \log d \cdot \left(\log d + \log L_1(\mathsf{Q}) + \log L_0(\mathsf{Q}) + \log \frac{1}{\varepsilon}\right)\right). \qquad (14)$$

In the typical setting where $\log d$ is (at most) logarithmic in $\log(1/\varepsilon)$, we can write

$$s(n, \varepsilon_0, \varepsilon) = O(s_0(n, \varepsilon_0)) + \widetilde{O}\left(\log\left(L_1(\mathsf{Q})L_0(\mathsf{Q})\right) + \log \frac{1}{\varepsilon}\right).$$

[7] We skip the details of how to compute the WPRG itself, based on Q, G_0, and G_{aux}. Recall that given a seed, we need to output an *instruction* in Σ^n, together with the instruction's *weight*. The complete details appear in [PV21, CDR+21] for $\mathsf{Q} = \mathsf{Rich}$, but a similar analysis extends to an arbitrary Q.

[8] The approach in [Hoz21] can be used to eliminate the doubly-logarithmic factors, which in our case amounts to the $\log d$ multiplicative factor.

The full details can be found in [PV21, CDR+21].

Let us illustrate how this framework is employed in [PV21, CDR+21] for ROBPs, and in [PV21] for permutation ROBPs. For *arbitrary* ROBPs, we take Nisan's generator as G_0, and use Rich_k as the error-reduction polynomial for $k = O(t) = O(d)$, and we have that $\varepsilon = n \cdot (n \cdot \varepsilon_0)^{t+1}$, so $t = O\left(\frac{\log(n/\varepsilon)}{\log(1/(n\varepsilon_0))}\right)$. Indeed, this can be inferred both from a simple L_1 bound, as above, or by employing the "optimization-based" analysis that was described in Section 1.2. Then, choosing $\varepsilon_0 \approx \frac{1}{n}$, and recalling that $L_1(\mathsf{Q}), L_0(\mathsf{Q}) = n^{O(k)}$, we get a WPRG with seed length $s = O(\log^2 n) + \widetilde{O}(\log(1/\varepsilon))$. Observe that it is indeed a WPRG and not a PRG, since inherently, some $c_{\mathbf{I}}$-s are negative.

For *permutation* branching programs, one can start with $s_0(n, \varepsilon_0) = O(\log n \cdot \log(1/\varepsilon_0))$.[9] Although [PV21] still uses Rich as the error reduction polynomial, they show by a sophisticated argument (following [AKM+20]), that the error ε satisfies $\varepsilon \approx n \cdot (\log n \cdot \varepsilon_0)^{t+1}$ under a suitable choice of norm. And so, in the error reduction step, they obtain an improvement over the L_1-norm based analysis. Plugging-in parameters, the resulting WPRG has seed of length

$$s = \widetilde{O}\left(\log n \cdot \log(1/\varepsilon_0) + \frac{\log(n/\varepsilon)}{\log(1/\varepsilon_0)} \cdot \log n + \log \frac{1}{\varepsilon}\right). \tag{15}$$

Balancing parameters, we see that the best choice of parameters becomes $\varepsilon_0 = 2^{-\sqrt{\log(1/\varepsilon)}}$, and the seed length amounts to $s = \widetilde{O}(\log n \cdot \sqrt{\log(1/\varepsilon)} + \log(1/\varepsilon))$. In particular, for error $\varepsilon = \frac{1}{\mathrm{poly}(n)}$, we get $s = \widetilde{O}(\log^{3/2} n)$. A similar balancing of parameters appears in the more recent work of [CHL+23], which will be discussed soon, when we talk about the potential benefit of working with correlated errors.

To conclude this section, we make the following important observations:

- Even when the L_1-norm does not play a role in the *error reduction* itself, say in the case for permutation and regular branching programs, it does manifest itself when employing it to get a WPRG, as evident from Equation (15), where we still had to take $L_1(\mathsf{Q}) = n^{\Omega(k)}$.[10]

- If d were extremely small, and we could still keep t sufficiently large, one can consider not using G_{aux}, which would lead to seed length $O(d \cdot s_0(n, \varepsilon_0) + \log L_0(\mathsf{Q}))$ which would have had interesting implications.

[9] For $w = O(1)$, this follows from the PRGs of [KNP11, De11, Ste12]. When w is non-constant, a better seed can be obtained by using the PRGs in [BRRY14, HPV21].

[10] Indeed, in the non black-box setting, when we're not aiming for a PRG, [AKM+20] obtains a high-precision algorithm that approximates random walks over Eulerian directed graphs and runs in space $\widetilde{O}(\log n)$.

The bounds on d, $L_1(\mathsf{Q})$, and $L_0(\mathsf{Q})$ from Theorem 2.1 imply that we *need* to use G_{aux} (namely, that the second step of the error reduction procedure is necessary), and that even a very good $s_0(n, \varepsilon_0)$ will not lead to a near-optimal dependence on ε (as apparent from Equation (14)).

2.4 Revisiting Our Underlying Assumption: Correlated Errors

In this section, we reassess the foundational assumption of our abstract framework concerning the independence of errors. This assumption proves to be excessively pessimistic since correlations between errors *can* indeed be introduced. Notably, such correlations already manifest in existing constructions as a natural consequence of the recursive structure of Nisan's PRG, and were exploited in error reduction procedures by Chen, Hoza, Lyu, Tal, and Wu [CHL+23] for width-3 and for bounded-width regular ROBPs.

To illustrate the role of correlations within our abstract framework, let us reconsider Equation (1) and focus on the term $\widetilde{A^3}$. Due to Nisan's recursive setup, this entry of the matrix is constructed either by combining $\widetilde{A^2}A$ or $A\widetilde{A^2}$, depending on the implementation. More broadly, $\widetilde{A^k}$ is generated by multiplying approximations of the powers-of-two of A that align with the bits in the binary representation of k. For instance, $\widetilde{A^{11}}$ is the product of the matrices $A, \widetilde{A^2}$, and $\widetilde{A^8}$, in varying orders.

This structure leads to a scenario where the independently generated error terms are $\boldsymbol{\varepsilon}_{2^i}$ for $i = 1, 2, ..., \lfloor \log_2 n \rfloor$. The remaining, vast majority of the error terms, however, are derived from these and from the number a that we aim to approximate, thereby introducing dependencies. For example:

$$a^3 + \boldsymbol{\varepsilon}_3 = \widetilde{a^3} = a\widetilde{a^2} = a(a^2 + \boldsymbol{\varepsilon}_2) = a^3 + a\boldsymbol{\varepsilon}_2,$$

thus, $\boldsymbol{\varepsilon}_3 = a\boldsymbol{\varepsilon}_2$. Similar calculations reveal that $\boldsymbol{\varepsilon}_5 = a\boldsymbol{\varepsilon}_4$. However, the dependencies gradually become more complex, e.g., $\boldsymbol{\varepsilon}_6 = a^2\boldsymbol{\varepsilon}_4 + a^4\boldsymbol{\varepsilon}_2 + \boldsymbol{\varepsilon}_2\boldsymbol{\varepsilon}_4$.

These correlations effectively reduce the number of independent error terms from n to $\log n$. This raises an important question: do our impossibility results, given by Theorem 2.1, still apply in the presence of correlations? At first it may appear that the answer is no, as the correlations can facilitate the development of error-reduction procedures that are unattainable under the assumption of independent errors. To illustrate this point, consider the fact that no error-reduction polynomial of degree-2 exists when assuming independent errors. However, this is not the case when errors are correlated.

To see this, let's revisit our initial example where $n = 4$, but now with $d = 2$. The general form of the polynomial is given by

$$A\boldsymbol{x}_1\boldsymbol{x}_3 + B\boldsymbol{x}_2^2, \tag{16}$$

where again we omit the monomial $\boldsymbol{x}_4$ with hindsight. Under the assumption of independent errors, such a polynomial cannot eliminates the linear error terms because the inevitable homogeneous constraints $A = B = 0$ lead to the zero polynomial. However, with the introduction of correlations as discussed earlier, the algebraic relation $\boldsymbol{x}_3 = \boldsymbol{x}_1\boldsymbol{x}_2$ emerges within our revised abstract framework. Consequently, Equation (16) transforms into

$$\begin{aligned} A\boldsymbol{x}_1^2\boldsymbol{x}_2 + B\boldsymbol{x}_2^2 &= Aa^2(a^2 + \boldsymbol{\varepsilon}_2) + B(a^2 + \boldsymbol{\varepsilon}_2)^2 \\ &= (A+B)a^4 + (A+2B)a^2\boldsymbol{\varepsilon}_2 + B\boldsymbol{\varepsilon}_2^2. \end{aligned}$$

This allows for the elimination of the linear error term. It is achieved by setting $A = 2$ and $B = -1$, resulting in the polynomial $2\boldsymbol{x}_1\boldsymbol{x}_3 - \boldsymbol{x}_2^2$.

The above may falsely lead to the conclusion that correlated errors allow for strictly more error-reduction polynomials. However, this is not the case. The complex nature of the correlations leads to some unexpected effects due to "monomial collapse", which arises from the algebraic relations induced by these correlations. Consider again the case of degree $d = 2$ polynomials, now with $n = 5$. The relevant monomials are $\boldsymbol{x}_1\boldsymbol{x}_4$, $\boldsymbol{x}_2\boldsymbol{x}_3$, and $\boldsymbol{x}_5$. However, the latter monomial, $\boldsymbol{x}_5$, collapses into the first monomial $\boldsymbol{x}_1\boldsymbol{x}_4$, resulting in a loss of one degree of freedom. This leaves us with a polynomial of the form

$$\begin{aligned} A\boldsymbol{x}_1\boldsymbol{x}_4 + B\boldsymbol{x}_2\boldsymbol{x}_3 &= A\boldsymbol{x}_1\boldsymbol{x}_4 + B\boldsymbol{x}_1\boldsymbol{x}_2^2 \\ &= Aa(a^4 + \boldsymbol{\varepsilon}_4) + Ba(a^2 + \boldsymbol{\varepsilon}_2)^2 \\ &= (A+B)a^5 + 2Ba^3\boldsymbol{\varepsilon}_2 + Aa\boldsymbol{\varepsilon}_4 + Ba\boldsymbol{\varepsilon}_2^2, \end{aligned}$$

from which it is evident that eliminating the linear error terms requires setting $A = B = 0$, contradicting the affine constraint $A + B = 1$. For $n = 7$, the collapses become even more significant, leaving us with only a single monomial, $\boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_4$. As a result, no error reduction procedure exists. This phenomenon occurs for every n of the form $n = 2^k - 1$.

On the other hand, to further appreciate the power of the uncorrelated errors setting, we note that in the latter it is possible to construct low degree polynomials in which also the quadratic error terms vanish. The smallest such example is when $n = 6$ and the degree $d = 4$. In this case, the polynomial $3\boldsymbol{x}_1^2\boldsymbol{x}_3 - 3\boldsymbol{x}_3^2 + \boldsymbol{x}_2^3$ when expressed in terms of a and $\boldsymbol{\varepsilon}_2$ simplifies to $a^6 + \boldsymbol{\varepsilon}_2^3$. One requires significantly higher degrees, as well as a higher value of n, to eliminate the quadratic terms in the non-correlated setting.

In summary, although the number of independent error terms in the correlated setting is significantly reduced from n to $\log n$, resulting in only approximately $(\log n)^t$ linear constraints rather than n^t, the degrees of freedom are also reduced. More importantly, the structure of these degrees of freedom remains elusive, making it more difficult to argue about in general.

Another way to view the correlated setting is as a mechanism that effectively "buys" a factor of approximately $\log n$ in the degree, albeit in a complex manner. For instance, the degree-1 monomial $\boldsymbol{x}_{15}$ is equivalent to the degree-4 monomial $\boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_4\boldsymbol{x}_8$. Generally, a variable $\boldsymbol{x}_k$ is equivalent to a monomial whose degree is the count of '1's in the binary representation of k. This $\log n$ factor would have had far-reaching consequences within the context of space-bounded derandomization, as can be inferred from Section 2.3. Nonetheless, control over this aspect is limited due to the intricate nature of the correlations. Given the above discussion and examples, it is not a priory clear whether correlated errors can lead to more effective error-reduction procedures.

As suggested in earlier sections, our second main result is a negative answer to this question. Somewhat surprisingly, we establish bounds that are comparable to those derived in Theorem 2.1 for the correlated setting as well, the only difference being a slightly weaker bound on the degree (see Theorem 4.2). Nonetheless, the proofs turned out to be more complex, requiring substantially more work

To conclude, we note that although our focus is on correlated errors specifically in settings similar to Nisan's—or more broadly, in recursive structures that reduce from length n to $\frac{n}{2}$—we believe that our ideas can be generalized to any reasonable extensions derived through recursive constructions.

A simultaneous bound. Finally, in Section 7, we address the feasibility of the following scenario: What if the large L_1 bound is concentrated only on monomials of small degree, whereas restricted to the high-degree monomials, the L_1 norm is small? This could conceivably offer a way to bypass the barriers towards constructing better WPRGs, presented in Section 2.3. Unfortunately, this cannot be the case. We defer the formal statement to Theorem 7.1.

3 Setting the Stage

We make use of the following notation and definitions. Unless stated otherwise, all logarithms in this paper are taken to the base 2. The set of natural numbers is $\mathbb{N} = \{0, 1, 2, \ldots\}$. For $n \in \mathbb{N}$, $n \geq 1$, we use $[n]$ to denote the set $\{1, \ldots, n\}$. For a vector $\mathbf{I} = (i_1, \ldots, i_n)$ of non-negative integers, we denote $|\mathbf{I}| = \sum_{k=1}^{n} i_k$ and $w(\mathbf{I}) = \sum_{k=1}^{n} k \cdot i_k$. For $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, let $R[\boldsymbol{x}] = R[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ be the ring of polynomials in variables $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ and coefficients in the ring R. For $\mathsf{Q} \in R[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ we use $\deg(\mathsf{Q})$ to denote the total degree of Q, and $\deg_{x_i}(\mathsf{Q})$ the denote the individual degree in x_i. We will use $\boldsymbol{x}^{\mathbf{I}}$ to denote the monomial $\prod_{k=1}^{n} \boldsymbol{x}_k^{i_k} \in R[\boldsymbol{x}]$.

Definition 3.1 (error reduction polynomial; uncorrelated errors)**.** Let $n, d, t \in \mathbb{N}$ be such that $d < n$. We say that a polynomial $\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in \mathbb{R}[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ is a (d,t)*-error reduction polynomial (for uncorrelated errors)* if Q is of total degree at most d, and there exist polynomials $\{c_{\mathbf{I}}(\boldsymbol{a})\}_{\mathbf{I}=(i_2,\ldots,i_n)\in\mathbb{N}^{n-1}} \subseteq \mathbb{R}[\boldsymbol{a}]$ such that

$$\mathsf{Q}\left(\boldsymbol{a}, \boldsymbol{a}^2 + \boldsymbol{\varepsilon}_2, \ldots, \boldsymbol{a}^n + \boldsymbol{\varepsilon}_n\right) = \boldsymbol{a}^n + \sum_{\substack{\mathbf{I}=(i_2,\ldots,i_n)\in\mathbb{N}^{n-1} \\ |\mathbf{I}|>t}} c_{\mathbf{I}}(\boldsymbol{a})\, \boldsymbol{\varepsilon}^{\mathbf{I}}\,, \tag{17}$$

where $\boldsymbol{\varepsilon} = (\boldsymbol{\varepsilon}_2, \ldots, \boldsymbol{\varepsilon}_n)$ and the equality is in the polynomial ring $\mathbb{R}[\boldsymbol{a}, \boldsymbol{\varepsilon}_2, \ldots, \boldsymbol{\varepsilon}_n]$.

Remark 3.2. We require that $d < n$ since otherwise for $d \geq n$ the polynomial $\boldsymbol{x}_1^n$ would be a trivial (d,t)-error reduction polynomial for every $t \in \mathbb{N}$.

Definition 3.3 (error reduction polynomial; correlated errors)**.** Let $n, d, t \in \mathbb{N}$ be such that $d < n$ and set $s = \lfloor \log n \rfloor$. We say that a polynomial $\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in \mathbb{R}[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ is a (d,t)*-error reduction polynomial for correlated errors* if its total-degree is at most d, and there exist polynomials $\{c_{\mathbf{I}}(\boldsymbol{a})\}_{\mathbf{I}=(i_1,\ldots,i_s)\in\mathbb{N}^{s}} \subseteq \mathbb{R}[\boldsymbol{a}]$ such that the following equality holds over $\mathbb{R}[\boldsymbol{a}, \boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_s]$:

$$\mathsf{Q}\left(\widetilde{\boldsymbol{a}^1}, \widetilde{\boldsymbol{a}^2}, \widetilde{\boldsymbol{a}^3}, \ldots, \widetilde{\boldsymbol{a}^n}\right) = \boldsymbol{a}^n + \sum_{\substack{\mathbf{I}=(i_1,\ldots,i_s)\in\mathbb{N}^{s} \\ |\mathbf{I}|>t}} c_{\mathbf{I}}(\boldsymbol{a})\, \boldsymbol{\varepsilon}^{\mathbf{I}}\,, \tag{18}$$

where $\boldsymbol{\varepsilon} = (\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_s)$ and for every $r \in [n]$ such that $r = \sum_{i=0}^{s} 2^i r_i$ for $r_i \in \{0,1\}$,

$$\widetilde{\boldsymbol{a}^r} = \boldsymbol{a}^{r_0} \prod_{i=1}^{s} \left(\boldsymbol{a}^{2^i} + \boldsymbol{\varepsilon}_i\right)^{r_i} \in \mathbb{R}[\boldsymbol{a}, \boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_s].$$

We say that a polynomial $\mathsf{Q}(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_s) \in \mathbb{R}[\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_s]$ is a *reduced* (d,t)*-error reduction polynomial (for correlated errors)* if each of its individual degrees is at most d, and there exist polynomials $\{c'_{\mathbf{I}}(\boldsymbol{a})\}_{\mathbf{I}\in\mathbb{N}^{s}} \subseteq \mathbb{R}[\boldsymbol{a}]$ such that the following equality holds over $\mathbb{R}[\boldsymbol{a}, \boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_s]$:

$$\mathsf{Q}\left(\widetilde{\boldsymbol{a}^1}, \widetilde{\boldsymbol{a}^2}, \widetilde{\boldsymbol{a}^4}, \ldots, \widetilde{\boldsymbol{a}^{2^s}}\right) = \boldsymbol{a}^n + \sum_{\substack{\mathbf{I}\in\mathbb{N}^{s} \\ |\mathbf{I}|>t}} c'_{\mathbf{I}}(\boldsymbol{a})\, \boldsymbol{\varepsilon}^{\mathbf{I}}\,, \tag{19}$$

where $\boldsymbol{\varepsilon} = (\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_s)$, $\widetilde{\boldsymbol{a}^1} = \boldsymbol{a}$, and for every $i \in [s]$

$$\widetilde{\boldsymbol{a}^{2^i}} = \boldsymbol{a}^{2^i} + \boldsymbol{\varepsilon}_i \in \mathbb{R}[\boldsymbol{a}, \boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_s].$$

Claim 3.4. *For every (d,t)-error reduction polynomial for correlated errors $\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ there exists a polynomial $\mathsf{Q}'(\boldsymbol{x}'_0, \boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_s)$ which is a reduced (d,t)-error reduction polynomial for correlated errors.*

Moreover, each monomial $\prod_{j=0}^{s} (\boldsymbol{x}'_j)^{v_j}$ which appears in Q' satisfies $\sum_{j=0}^{s} 2^j \cdot v_j = w(\mathbf{I})$ for some $\mathbf{I} \in \mathbb{N}^n$ such that $\boldsymbol{x}^{\mathbf{I}}$ appears in Q.

Proof. For every monomial $\boldsymbol{x}^{\mathbf{I}}$ for $\mathbf{I} \in \mathbb{N}^n$ which appears in Q we will have an equivalent monomial of Q'. The conversion is done by using the rule $\boldsymbol{x}_r = \prod_{i \in \ \{0,\ldots,s\} | r_i = 1} \boldsymbol{x}'_i$ where for every $r \in [n]$ $r_0, \ldots, r_s \in \{0,1\}$ are the binary representation of r, $r = \sum_{i=0}^{s} 2^i r_i$. If we convert every monomial $\boldsymbol{x}^{\mathbf{I}}$ of Q according to this rule, every variable $\boldsymbol{x}'_i$ for $i \in \{0, \ldots, s\}$ will have individual degree at most d since $\boldsymbol{x}'_i$ appears at most once when substituting each of the variables of $\boldsymbol{x}^{\mathbf{I}}$. It is immediate that if Q satisfied Equation (18) then Q' satisfies Equation (19). As for the moreover part of the claim, one can observe that the described conversion transforms each monomial $\boldsymbol{x}^{\mathbf{I}}$ in Q to a monomial of the form $\prod_{j=0}^{s} (\boldsymbol{x}'_j)^{v_j}$ with $\sum_{j=0}^{s} 2^j \cdot v_j = w(\mathbf{I})$. □

Occasionally, it may be convenient to assume that all the monomials $\boldsymbol{x}^{\mathbf{I}}$ which appear in an error-reduction polynomial $\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ satisfy $w(\mathbf{I}) = n$. To facilitate this, the following lemma will prove to be useful.

Lemma 3.5. *Let $\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{v \in \mathbb{N}^n} \gamma_v \boldsymbol{x}^v$ be a (d,t)-error reduction polynomial for uncorrelated (resp. correlated) errors. Then, the polynomial*

$$\mathsf{Q}'(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{\substack{v \in \mathbb{N}^n \\ w(v) = n}} \gamma_v \boldsymbol{x}^v$$

is also a (d,t)-error reduction polynomial for uncorrelated (resp. correlated) errors.

Proof. Set $s = \lfloor \log n \rfloor$. We will prove the lemma by establishing the following fact. For any polynomial $\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ if

$$\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{v \in \mathbb{N}^n} \gamma_v \boldsymbol{x}^v, \tag{20}$$

satisfies Equation (17), then

$$\mathsf{Q}'(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{v \in \mathbb{N}^n | w(v) = n} \gamma_v \boldsymbol{x}^v \tag{21}$$

also satisfies Equation (17). Before we turn to prove this fact, we argue that indeed it implies the lemma. First, in the case of uncorrelated errors the implication is trivial, since clearly if Q is of total degree at most d then so is Q', and we have that like Q, Q' also satisfies Equation (17) – and so Q' is a (d,t)-error reduction polynomial.

Secondly, we argue that in the case of correlated errors the implication holds as well. This follows by observing that a polynomial $\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ satisfies Equation (18) if and only if the polynomial $\widehat{\mathsf{Q}}(\widehat{\boldsymbol{x}}_1, \ldots, \widehat{\boldsymbol{x}}_n)$ satisfies Equation (17), where the polynomial $\widehat{\mathsf{Q}}(\widehat{\boldsymbol{x}}_1, \ldots, \widehat{\boldsymbol{x}}_n)$ is obtained from $\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ by substituting every variable $\boldsymbol{x}_r$, for $r \in [n]$, which appears in Q, with the product $\prod_{i \in \ \{0,\ldots,s\} | r_i = 1} \widehat{\boldsymbol{x}}_{2^i}$ where $r_0, \ldots, r_s \in \{0,1\}$ are the binary representation of r, $r = \sum_{i=0}^{s} 2^i r_i$. The observation is immediate by inspecting Equation (18). Thus, if $\mathsf{Q}(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)$ is a (d,t)-error reduction polynomial for correlated errors then it satisfies Equation (18), and so $\widehat{\mathsf{Q}}(\widehat{\boldsymbol{x}}_1,\dots,\widehat{\boldsymbol{x}}_n)$ satisfies Equation (17). Now by the assumed fact, we have that $(\widehat{\mathsf{Q}})'(\widehat{\boldsymbol{x}}_1,\dots,\widehat{\boldsymbol{x}}_n)$ also satisfies Equation (17). Note that $(\widehat{\mathsf{Q}})' = \widehat{(\mathsf{Q}')}$ as the substitution rule defining $\widehat{\mathsf{Q}}$ maintains the weight $w(v)$ of a monomial $\boldsymbol{x}^v$. Hence, equivalently, we have that $\widehat{(\mathsf{Q}')}$ satisfies Equation (17). Therefore, again by the observation, $\mathsf{Q}'(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)$ satisfies Equation (18), and so it is a (d,t)-error reduction polynomial for correlated errors.

Now, we can proceed to proving the claimed fact. Towards that, we assume that Q satisfies Equation (17), and we write

$$\mathsf{Q}'(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n) = \sum_{v\in\mathbb{N}^n : w(v)=n} \gamma_v \boldsymbol{x}^v = \sum_{v\in\mathbb{N}^n} \gamma'_v \boldsymbol{x}^v, \tag{22}$$

where for every v, $\gamma'_v = \gamma_v$ if $w(v) = n$, and $\gamma'_v = 0$ otherwise. In order to show that Q' also satisfies Equation (17), let us first define two polynomials $\mathsf{P}, \mathsf{P}' \in \mathbb{R}[\boldsymbol{a}, \boldsymbol{\varepsilon}_2,\dots,\boldsymbol{\varepsilon}_n]$ by

$$\mathsf{P} \triangleq \mathsf{Q}(\boldsymbol{a}, \boldsymbol{a}^2+\boldsymbol{\varepsilon}_2,\dots,\boldsymbol{a}^n+\boldsymbol{\varepsilon}_n) = \sum_{v\in\mathbb{N}^n} \gamma_v\, \boldsymbol{a}^{v_1} \prod_{i=2}^{n} (\boldsymbol{a}^i+\boldsymbol{\varepsilon}_i)^{v_i}, \tag{23}$$

and similarly

$$\mathsf{P}' \triangleq \mathsf{Q}'(\boldsymbol{a}, \boldsymbol{a}^2+\boldsymbol{\varepsilon}_2,\dots,\boldsymbol{a}^n+\boldsymbol{\varepsilon}_n) = \sum_{v\in\mathbb{N}^n} \gamma'_v\, \boldsymbol{a}^{v_1} \prod_{i=2}^{n} (\boldsymbol{a}^i+\boldsymbol{\varepsilon}_i)^{v_i}. \tag{24}$$

To simplify notation, through the remaining part we will use $[M]_f$ to denote the coefficient of the monomial M in a polynomial f. As Q satisfies Equation (17) we have that for every $j \in \mathbb{N}$ and $\mathbf{I} = (I_2,\dots,I_n) \in \mathbb{N}^{n-1}$ such that $|\mathbf{I}| \le t$,

$$\left[\boldsymbol{a}^j \boldsymbol{\varepsilon}^{\mathbf{I}}\right]_{\mathsf{P}} = \begin{cases} 1 & j = n \text{ and } \mathbf{I} = 0; \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, a simple inspection of the RHS of Equation (23) and Equation (24) yields

$$[\boldsymbol{a}^n]_{\mathsf{P}'} = \sum_{\substack{v\in\mathbb{N}^n,\\ w(v)=n}} \gamma'_v = \sum_{\substack{v\in\mathbb{N}^n,\\ w(v)=n}} \gamma_v = [\boldsymbol{a}^n]_{\mathsf{P}} = 1, \tag{25}$$

and that for every $j \neq n$,

$$\left[\boldsymbol{a}^j\right]_{\mathsf{P}'} = \sum_{\substack{v\in\mathbb{N}^n,\\ w(v)=j}} \gamma'_v = \sum_{\substack{v\in\mathbb{N}^n,\\ w(v)=j}} 0 = 0.$$

Furthermore, for every $j \in \mathbb{N}$ and $\mathbf{I} = (I_2, \ldots, I_n) \in \mathbb{N}^{n-1}$ such that $0 < |\mathbf{I}| \leq t$,

$$\left[\boldsymbol{a}^j \boldsymbol{\varepsilon}^{\mathbf{I}}\right]_{\mathsf{P}'} = \sum_{\substack{v \in \mathbb{N}^n, \\ w(v) = j + \sum_{i=2}^n iI_i}} \gamma'_v \prod_{i=2}^n \binom{v_i}{I_i}.$$

Thus, if j and $\mathbf{I}$ satisfy in addition that $j + \sum_{i=2}^n iI_i = n$, then

$$\left[\boldsymbol{a}^j \boldsymbol{\varepsilon}^{\mathbf{I}}\right]_{\mathsf{P}'} = \sum_{\substack{v \in \mathbb{N}^n, \\ w(v) = n}} \gamma'_v \prod_{i=2}^n \binom{v_i}{I_i} = \sum_{\substack{v \in \mathbb{N}^n, \\ w(v) = j + \sum_{i=2}^n iI_i}} \gamma_v \prod_{i=2}^n \binom{v_i}{I_i} = \left[\boldsymbol{a}^j \boldsymbol{\varepsilon}^{\mathbf{I}}\right]_{\mathsf{P}} = 0. \tag{26}$$

Otherwise,

$$\left[\boldsymbol{a}^j \boldsymbol{\varepsilon}^{\mathbf{I}}\right]_{\mathsf{P}'} = \sum_{\substack{v \in \mathbb{N}^n, \\ w(v) = j + \sum_{i=2}^n iI_i}} \gamma'_v \prod_{i=2}^n \binom{v_i}{I_i} = \sum_{\substack{v \in \mathbb{N}^n, \\ w(v) = j + \sum_{i=2}^n iI_i}} 0 \cdot \prod_{i=2}^n \binom{v_i}{I_i} = 0. \tag{27}$$

In summary, for every $j \in \mathbb{N}$ and $\mathbf{I} = (I_2, \ldots, I_n) \in \mathbb{N}^n$ such that $|\mathbf{I}| \leq t$, we have

$$\left[\boldsymbol{a}^j \boldsymbol{\varepsilon}^{\mathbf{I}}\right]_{\mathsf{P}'} = \begin{cases} 1 & j = n \text{ and } \mathbf{I} = 0; \\ 0 & \text{otherwise.} \end{cases}$$

From which we conclude that Q' satisfies Equation (17). The fact, and the claim, follow. □

4 Degree Bound

In this section, we establish our lower bounds for the degree of error-reduction polynomials. The uncorrelated setting, which we cover next, has a simple and straightforward proof. In contrast, the correlated setting, discussed in Section 4.1, proves to be more challenging.

Proposition 4.1 (degree bound, uncorrelated errors)**.** *Let $n, d, t \in \mathbb{N}$ and let $\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ be a (d,t)-error reduction polynomial for uncorrelated errors. Then, $d > t$.*

Proof. Assume for the sake of contradiction that $d \leq t$. Since Q is a (d,t)-error reduction polynomial, there exist $\{c_{\mathbf{I}}(\boldsymbol{a})\}_{\mathbf{I}=(i_2,\ldots,i_n)\in\mathbb{N}^{n-1}} \subseteq \mathbb{R}[\boldsymbol{a}]$ such that

$$\mathsf{Q}\left(\boldsymbol{a}, \boldsymbol{a}^2 + \boldsymbol{\varepsilon}_2, \ldots, \boldsymbol{a}^n + \boldsymbol{\varepsilon}_n\right) = \boldsymbol{a}^n + \sum_{\substack{\mathbf{I}=(i_2,\ldots,i_n)\in\mathbb{N}^{n-1} \\ |\mathbf{I}|>t}} c_{\mathbf{I}}(\boldsymbol{a})\, \boldsymbol{\varepsilon}^{\mathbf{I}},$$

where $\boldsymbol{\varepsilon} = (\boldsymbol{\varepsilon}_2, \ldots, \boldsymbol{\varepsilon}_n)$. It must be that

$$\sum_{\substack{\mathbf{I}=(i_2,\ldots,i_n)\in\mathbb{N}^{n-1} \\ |\mathbf{I}|>t}} c_{\mathbf{I}}(\boldsymbol{a})\, \boldsymbol{\varepsilon}^{\mathbf{I}} = 0$$

as otherwise we would have that $d \geq \deg(\mathsf{Q}) > t$, contrary to our assumption. Thus,

$$\mathsf{Q}\left(\boldsymbol{a}, \boldsymbol{a}^2 + \boldsymbol{\varepsilon}_2, \ldots, \boldsymbol{a}^n + \boldsymbol{\varepsilon}_n\right) = \boldsymbol{a}^n.$$

Substituting $\boldsymbol{\varepsilon}_i = -\boldsymbol{a}^i$ for all $2 \leq i \leq n$, we get that $\mathsf{Q}(\boldsymbol{a}, 0, \ldots, 0) = \boldsymbol{a}^n$ in the ring $\mathbb{R}[\boldsymbol{a}]$. But this contradicts the fact that

$$n = \deg \mathsf{Q}(\boldsymbol{a}, 0, \ldots, 0) \leq \deg \mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \leq d < n,$$

where the last inequality follows by Definition 3.1. □

4.1 Correlated Errors

Theorem 4.2 (degree bound, correlated errors)**.** *Let $n, d, t \in \mathbb{N}$ be such that $t < n$ and let $\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in \mathbb{R}[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ be a (d,t)-error reduction polynomial for correlated errors. Then, $d > t/2$.*

Proof. Let $s = \lfloor \log n \rfloor$ and $D = \{0, \ldots, d\}$. We start by observing that by Lemma 3.5 we can assume without loss of generality that every monomial $\boldsymbol{x}^v$ for $v \in \mathbb{N}^n$, supported on $\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, satisfies $w(v) = n$. We further observe that by Claim 3.4, it is enough to show that for $\mathsf{Q}'(\boldsymbol{y}_0, \ldots, \boldsymbol{y}_s)$ a reduced (d,t)-error reduction polynomial for correlated errors, satisfying that for every monomial $\boldsymbol{y}^u$ for $u = (u_0, \ldots, u_s) \in D^{s+1}$ supported on Q'

$$\sum_{i=0}^{s} 2^i u_i = n, \tag{28}$$

it must be that $d > t/2$. We therefore proceed to show $d > t/2$ assuming such Q'.

Towards that we first write Q' as a polynomial arranged by the set of "high order" monomials

$$\left\{ \prod_{i=w}^{s} \boldsymbol{y}_i^{u_i} : u_w, \ldots, u_s \in D \right\}$$

for $w = \lceil \log(n/t) \rceil$. That is, viewing Q' as an element of the ring

$$\left(\mathbb{R}\left[\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{w-1}\right]\right)\left[\boldsymbol{y}_w, \ldots, \boldsymbol{y}_s\right],$$

we write

$$\mathsf{Q}' = \sum_{u_w, \ldots, u_s \in D} \mathsf{Q}_{u_w, \ldots, u_s}(\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{w-1}) \prod_{i=w}^{s} \boldsymbol{y}_i^{u_i}.$$

where for every $u_w, \ldots, u_s$, $\mathsf{Q}_{u_w,\ldots,u_s} \in \mathbb{R}[\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{w-1}]$ is a polynomial of individual degree at most d. We observe that for every $u_w, \ldots, u_s$ such that $\mathsf{Q}_{u_w,\ldots,u_s} \neq 0$ it must be that

$$\sum_{i=w}^{s} u_i \leq t. \tag{29}$$

Indeed, by Equation (28), $\sum_{i=w}^{s} 2^i u_i \leq n$, and in particular,

$$2^{\log(n/t)} \sum_{i=\lceil \log(n/t) \rceil}^{s} u_i = \frac{n}{t} \sum_{i=\lceil \log(n/t) \rceil}^{s} u_i \leq n.$$

The rest of the argument will use the properties of Q' under the assignment $\boldsymbol{y}_0 = \widetilde{\boldsymbol{a}^{2^0}}, \ldots, \boldsymbol{y}_s = \widetilde{\boldsymbol{a}^{2^s}}$, where recall that $\widetilde{\boldsymbol{a}^1} = \boldsymbol{a}$ and $\widetilde{\boldsymbol{a}^{2^i}} = \boldsymbol{a}^{2^i} + \boldsymbol{\varepsilon}_i$ for $i \in [s]$. Therefore, for ease of notation, we define

$$\mathsf{P} \triangleq \mathsf{Q}'\left(\widetilde{\boldsymbol{a}^1}, \widetilde{\boldsymbol{a}^2}, \widetilde{\boldsymbol{a}^4}, \ldots, \widetilde{\boldsymbol{a}^{2^s}}\right),$$
$$\mathsf{P}_{u_w,\ldots,u_s} \triangleq \mathsf{Q}_{u_w,\ldots,u_s}\left(\widetilde{\boldsymbol{a}^1}, \widetilde{\boldsymbol{a}^2}, \ldots, \widetilde{\boldsymbol{a}^{2^{w-1}}}\right),$$

where $\mathsf{P} \in \mathbb{R}[\boldsymbol{a}, \boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_s]$ and $\mathsf{P}_{u_w,\ldots,u_s} \in \mathbb{R}[\boldsymbol{a}, \boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_{w-1}]$ for every $u_w, \ldots, u_s \in D$. By Equation (19), P is only supported on monomials $\boldsymbol{a}^j \boldsymbol{\varepsilon}^{\mathbf{I}}$ such that if $|\mathbf{I}| \neq 0$ then

$$|\mathbf{I}| > t. \tag{30}$$

Next, we turn to inspect $\mathsf{P}_{u_w,\ldots,u_s}$ when assigning the "low order" errors to be zero, that is, when setting $\boldsymbol{\varepsilon}_1 = 0, \ldots, \boldsymbol{\varepsilon}_{w-1} = 0$.

Claim 4.3. *For every $(u_w, \ldots, u_s) \neq (0, \ldots, 0)$, it holds that*

$$\mathsf{P}_{u_w,\ldots,u_s}(\boldsymbol{a}, 0, \ldots, 0) = 0.$$

Proof. Assume for the sake of contradiction that the opposite is true. Let

$$(u_w^*, \ldots, u_s^*) \neq (0, \ldots, 0)$$

be a "maximal" tuple for which $\mathsf{P}_{u_w^*,\ldots,u_s^*}(\boldsymbol{a}, 0, \ldots, 0) \neq 0$. By maximal we mean that every *other* tuple $(u_w, \ldots, u_s)$, such that $u_w \geq u_w^* \wedge \cdots \wedge u_s \geq u_s^*$, satisfies $P_{u_w,\ldots,u_s}(\boldsymbol{a}, 0, \ldots, 0) = 0$ (in particular, $(u_w^*, \ldots, u_s^*) \neq (0, \ldots, 0)$). In this case, we have that the polynomial P under the assignment of the low order errors to zero

$$\mathsf{P}(\boldsymbol{a}, 0, \ldots, 0, \boldsymbol{\varepsilon}_w, \ldots, \boldsymbol{\varepsilon}_s) = \sum_{u_w,\ldots,u_s \in D} \mathsf{P}_{u_w,\ldots,u_s}(\boldsymbol{a}, 0, \ldots, 0) \prod_{i=w}^{s} (\boldsymbol{a}^{2^i} + \boldsymbol{\varepsilon}_i)^{u_i}$$

is supported on the error term $\prod_{i=w}^{s} \boldsymbol{\varepsilon}_i^{u_i^*}$. However, by our choice $\sum_{i=w}^{t} u_i^* > 0$, and by Equation (29), $\sum_{i=w}^{t} u_i^* \leq t$, this stands in contradiction to Equation (30). We remark in addition to that as we chose $(u_w^*, \ldots, u_s^*)$ to be maximal, it cannot be that the non-zero term

$$\mathsf{P}_{u_w^*,\ldots,u_s^*}(\boldsymbol{a}, 0, \ldots, 0) \prod_{i=w}^{s} \boldsymbol{\varepsilon}_i^{u_i^*}$$

coming from

$$\mathsf{P}_{u_w^*,\ldots,u_s^*}(\boldsymbol{a}, 0, \ldots, 0) \prod_{i=w}^{s} (\boldsymbol{a}^{2^i} + \boldsymbol{\varepsilon}_i)^{u_i^*} \boldsymbol{\varepsilon}_i^{u_i^*}$$

is canceled out by other terms of $\mathsf{P}(\boldsymbol{a}, 0, \ldots, 0, \boldsymbol{\varepsilon}_w, \ldots, \boldsymbol{\varepsilon}_s)$, since for other non-zero

$$\mathsf{P}_{u_w,\ldots,u_s}(\boldsymbol{a}, 0, \ldots, 0) \prod_{i=w}^{s} (\boldsymbol{a}^{2^i} + \boldsymbol{\varepsilon}_i)^{u_i}$$

it must be that for some i, $u_i < u_i^*$, and so all its monomials fall short from cancelling

$$\mathsf{P}_{u_w,\ldots,u_s}(\boldsymbol{a}, 0, \ldots, 0) \prod_{i=w}^{s} \boldsymbol{\varepsilon}_i^{u_i^*},$$

which completes the proof. □

As a final step, we proceed by further consider P under the assignment that sets *all* error terms to 0. On the one hand, we have that

$$\mathsf{P}(\boldsymbol{a}, 0, \ldots, 0) = \mathsf{Q}'(\boldsymbol{a}, \boldsymbol{a}^2, \ldots, \boldsymbol{a}^{2^s}) = \boldsymbol{a}^n,$$

by virtue of Q' being a (d, t)-error reduction polynomial. On the other hand, by Claim 4.3,

$$\begin{aligned}\mathsf{Q}'(\boldsymbol{a}, \boldsymbol{a}^2, \ldots, \boldsymbol{a}^{2^s}) &= \sum_{u_w,\ldots,u_s \in D} \mathsf{P}_{u_w,\ldots,u_s}(\boldsymbol{a}, 0, \ldots, 0) \prod_{i=w}^{s} \boldsymbol{a}^{2^i u_i} \\ &= \mathsf{P}_{0,\ldots,0}(\boldsymbol{a}, 0, \ldots, 0),\end{aligned}$$

and therefore

$$\deg(\mathsf{P}_{0,\ldots,0}(\boldsymbol{a}, 0, \ldots, 0)) = n.$$

Now, recall that

$$\mathsf{P}_{0,\ldots,0}(\boldsymbol{a}, 0, \ldots, 0) = \mathsf{Q}_{0,\ldots,0}\left(\boldsymbol{a}, \boldsymbol{a}^2, \ldots, \boldsymbol{a}^{2^{w-1}}\right),$$

where the individual degree of $\mathsf{Q}_{0,\dots,0}(\boldsymbol{y}_0,\dots,\boldsymbol{y}_{w-1})$ is at most d. Therefore,

$$\begin{aligned}\deg(\mathsf{P}_{0,\dots,0}(\boldsymbol{a},0,\dots,0)) &\le \sum_{i=0}^{w-1} 2^i \cdot d\\ &= \sum_{i=0}^{\lceil \log(n/t)\rceil-1} 2^i \cdot d\\ &= (2^{\lceil \log(n/t)\rceil}-1)d.\end{aligned}$$

Hence, it must be that

$$(2^{\lceil \log(n/t)\rceil}-1)d \ge n,$$

from which it readily follows that $d > t/2$, as desired. □

5 L_1 Norm Bound

In this section we show that a (d,t)-error reduction polynomial must have a large L_1-norm. Recall that the L_1-*norm* of a polynomial $\mathsf{Q} \in \mathbb{R}[\boldsymbol{x}_1,\dots,\boldsymbol{x}_n]$,

$$\mathsf{Q}(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n) = \sum_{(v_1,\dots,v_n)\in\mathbb{N}^n} c_{(v_1,\dots,v_n)}\boldsymbol{x}_1^{v_1}\cdots\boldsymbol{x}_n^{v_n}$$

is $\sum_{(v_1,\dots,v_n)\in\mathbb{N}^n} |c_{(v_1,\dots,v_n)}|$. For every $n \in \mathbb{N}$ and $d \in \mathbb{N}$, define

$$M(n,d) = \left\{(v_1,\dots,v_n)\in\mathbb{N}^n : v_1 \le d,\ \sum_{i=1}^n i\cdot v_i = n\right\}.$$

In words, $M(n,d)$ describes the set of (powers of) "monomials", where we will think of each such monomial as being $\boldsymbol{a}^{v_1}(\boldsymbol{a}^2+\boldsymbol{\varepsilon}_2)^{v_2}\cdots(\boldsymbol{a}^n+\boldsymbol{\varepsilon}_n)^{v_n} \in \mathbb{R}[\boldsymbol{a},\boldsymbol{\varepsilon}_2,\dots,\boldsymbol{\varepsilon}_n]$. When defining this set we only consider powers whose weighted sum is equal to n, and the power of $\boldsymbol{a}$ is at most d. We now start by defining a representation of a (d,t)-error reduction polynomial that will be useful for lower-bounding the L_1-norm.

Definition 5.1. We say that $\bar{\gamma} = (\gamma_m)_{m\in M(n,d)}$, where each $\gamma_m \in \mathbb{R}$ is a (d,t)-coefficient vector if

$$\sum_{m\in M(n,d)} \gamma_m = 1, \tag{31}$$

and for every $I = (I_2,\dots,I_n) \in \mathbb{N}^{n-1}$ such that $0 < \sum I_i \le t$, it holds that

$$\sum_{(v_1,\dots,v_n)\in M(n,d)} \gamma_{(v_1,\dots,v_n)} \prod_{i=2}^n \binom{v_i}{I_i} = 0. \tag{32}$$

The L_1-norm of $\bar{\gamma}$ is $\sum_{m\in M(n,d)} |\gamma_m|$.

Indeed, a (d,t)-error reduction polynomial implies a (d,t)-coefficient vector, as we have in the following claim.

Claim 5.2. *Let $\mathsf{Q}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)$ be a (d,t)-error reduction polynomial of L_1-norm ℓ, for uncorrelated or correlated errors. Then, there exists a (d,t)-coefficient vector with L_1-norm at most ℓ.*

Proof. For a set $M \subseteq M(n,d)$ and coefficients $\bar{\gamma} = (\gamma_v)_{v\in M}$, let $\bar{\gamma}_{M(n,d)}$ be the extension of $\bar{\gamma}$ to $M(n,d)$ defined by setting $\gamma_v = 0$ for $v \in M(n,d)\setminus M$. We start the proof by noting that if for a set $M \subseteq M(n,d)$ we have that

$$\mathsf{Q}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n) = \sum_{v\in M} \gamma_v \boldsymbol{x}^v$$

satisfies Equation (17), then $\bar{\gamma}_{M(n,d)}$ for $\bar{\gamma} = (\gamma_v)_{v\in M}$, is a (d,t)-coefficient vector. Indeed, the fact that $\bar{\gamma}_{M(n,d)}$ satisfies Equation (31) can be seen by considering the coefficient $[\boldsymbol{a}^n]_{\mathsf{P}}$ mentioned in Equation (25). Further, $\bar{\gamma}_{M(n,d)}$ satisfies Equation (32) for every $I = (I_2,\ldots,I_n) \in \mathbb{N}^{n-1}$ such that $0 < \sum I_i \le t$, and this follows by considering the coefficients mentioned in Equation (26) and Equation (27).

We start by proving the claim for the case of uncorrelated errors. We have that if Q is a (d,t)-error reduction polynomial for uncorrelated errors, then by Lemma 3.5, without loss of generality, it is of the form

$$\mathsf{Q}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n) = \sum_{\substack{v\in\mathbb{N}^n, |v|\le d \\ \sum iv_i = n}} \gamma_v \boldsymbol{x}^v.$$

Thus, in this case we have that $M = \{n \in \mathbb{N}^n : |v| \le d, \sum iv_i = n\}$ and $\bar{\gamma} = (\gamma_v)_{v\in M}$, and we can take $\bar{\gamma}_{M(n,d)}$ to be the desired (d,t)-coefficient vector. Notice that indeed we have $M \subseteq M(n,d)$ as $|v| \le d$ implies $v_1 \le d$ in particular.

The proof for correlated errors is similar albeit less direct. Assume that $\mathsf{Q}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)$ is a (d,t)-polynomial for correlated errors and is of the same form as above, however, now we do not have that Q satisfies Equation (17). Set $s = \lfloor \log n \rfloor$. Similarly to the beginning of the proof for Lemma 3.5, we consider the polynomial $\widehat{\mathsf{Q}}(\widehat{\boldsymbol{x}}_1,\ldots,\widehat{\boldsymbol{x}}_n)$, achieved from $\mathsf{Q}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)$ by substituting every variable $\boldsymbol{x}_r$, for $r \in [n]$, which appears in Q, with the product $\prod_{i\in\ \{0,\ldots,s\}:r_i=1} \widehat{\boldsymbol{x}}_{2^i}$ where $r_0,\ldots,r_s \in \{0,1\}$ are the binary representation of r, $r = \sum_{i=0}^{s} 2^i r_i$. As in the proof for Lemma 3.5, since $\mathsf{Q}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)$ satisfies Equation (18), $\widehat{\mathsf{Q}}(\widehat{\boldsymbol{x}}_1,\ldots,\widehat{\boldsymbol{x}}_n)$ satisfies Equation (17). Notice that $\widehat{\mathsf{Q}}(\widehat{\boldsymbol{x}}_1,\ldots,\widehat{\boldsymbol{x}}_n)$ is a polynomial of individual degree at most d, and like Q, is supported only on monomials of weight n (as mentioned in the proof for Lemma 3.5 the transformation from Q to $\widehat{\mathsf{Q}}$ preserves the weight of monomials), and it is easy to check that its L_1-norm is smaller or equal to that of Q. Therefore, we can define $M = \{n \in \mathbb{N}^n : \max(v_1, \ldots, v_n) \leq d, \sum iv_i = n\}$ and have that

$$\widehat{\mathsf{Q}}(\widehat{\boldsymbol{x}}_1, \ldots, \widehat{\boldsymbol{x}}_n) = \sum_{v \in M} \widehat{\gamma}_v \widehat{\boldsymbol{x}}^v$$

satisfies Equation (17), and $M \subseteq M(n,d)$ since $v \in M$ in particular implies $v_1 \leq d$. Therefore, by the account above, $\bar{\gamma}_{M(n,d)}$ for $\bar{\gamma} = (\widehat{\gamma}_v)_{v \in M}$ is a (d,t)-coefficient vector as desired, completing the proof. □

Before proving our main result for this section, we will give two auxiliary claims.

Claim 5.3. *The set of equations given in Equation (32) imply that for every $I = (I_2, \ldots, I_n) \in \mathbb{N}^{n-1}$ such that $0 < \sum I_i \leq t$,*

$$\sum_{(v_1,\ldots,v_n) \in M(n,d)} \gamma_{(v_1,\ldots,v_n)} \prod_{i=2}^{n} {v_i}^{I_i} = 0. \tag{33}$$

Proof. For every I such that $0 < \sum I_i \leq t$ and $i \in \{2, \ldots, n\}$ such that $I_i > 0$, there exist $\beta_1^{I_i}, \ldots, \beta_{I_i}^{I_i} \in \mathbb{R}$ such that for every x

$$x^{I_i} = \sum_{r=1}^{I_i} \beta_r^{I_i} \binom{x}{r}$$

since the basis $\{\binom{x}{1}, \ldots, \binom{x}{I_i}\}$ spans x^{I_i}. Hence, let us fix some i such that $I_i > 0$. Consider the expression obtained by taking the LHS of Equation (32) and replacing in it $\binom{v_{i'}}{I_{i'}}$ with ${v_{i'}}^{I_{i'}}$:

$$\sum_{(v_1,\ldots,v_n) \in M(n,d)} \gamma_{(v_0,\ldots,v_s)} {v_{i'}}^{I_{i'}} \prod_{i \in \{2,\ldots,n\} \setminus \{i'\}} \binom{v_i}{I_i} =$$

$$\sum_{(v_1,\ldots,v_n) \in M(n,d)} \gamma_{(v_1,\ldots,v_n)} \sum_{r=1}^{I_{i'}} \beta_r^{I_{i'}} \binom{v_{i'}}{r} \prod_{i \in \{2,\ldots,n\} \setminus \{i'\}} \binom{v_i}{I_i} =$$

$$\sum_{r=1}^{I_{i'}} \beta_r^{I_{i'}} \sum_{(v_1,\ldots,v_n) \in M(n,d)} \gamma_{(v_1,\ldots,v_n)} \binom{v_{i'}}{r} \prod_{i \in \{2,\ldots,n\} \setminus \{i'\}} \binom{v_i}{I_i}.$$

For every $r \in [I_{i'}]$, we have that

$$\sum_{(v_1,\ldots,v_n) \in M(n,d)} \gamma_{(v_1,\ldots,v_n)} \binom{v_{i'}}{r} \prod_{i \in \{2,\ldots,n\} \setminus \{i'\}} \binom{v_i}{I_i} = 0$$

by Equation (32) which corresponds to $I' = (I_2, \ldots, I_{i'-1}, r, I_{i'+1}, \ldots, I_n)$ (notice that $0 < \sum I'_i \leq t$ since $1 \leq r \leq I_{i'}$). Therefore,

$$\sum_{(v_1,\ldots,v_n)\in M(n,d)} \gamma_{(v_1,\ldots,v_n)} {v_{i'}}^{I_{i'}} \prod_{i\in\{2,\ldots,n\}\setminus\{i'\}} \binom{v_i}{I_i} = 0.$$

To conclude the claim we can continue, with the same process, replacing all remaining $\binom{v_i}{I_i}$ (for which $I_i > 0$) with $v_i^{I_i}$. □

Claim 5.4. *There exist $\{\delta_I \in \mathbb{R}\}_{I=(I_2,\ldots,I_n)\in\mathbb{N}^{n-1}, 0<\sum I_i \leq t}$ such that for every monomial $m = (v_1, \ldots, v_n) \in M(n,d)$*

$$\sum_{\substack{I=(I_2,\ldots,I_n)\in\mathbb{N}^{n-1}\\ 0<\sum I_i\leq t}} \delta_I v_n^{I_n} \cdots v_2^{I_2} = n^t - v_1^t.$$

Proof. Since $m \in M(n,d)$,

$$\left(v_1 + \sum_{i=2}^{n} 2^i v_i\right)^t = n^t.$$

Hence,

$$\sum_{h=0}^{t-1} \binom{t}{h} v_1^h \left(\sum_{i=2}^{n-1} 2^i v_i\right)^{t-h} = n^t - v_1^t.$$

Plugging $v_1 = n - \sum_{i=2}^{n-1} 2^i v_i$,

$$\sum_{h=0}^{t-1} \binom{t}{h} \left(n - \sum_{i=2}^{n-1} 2^i v_i\right)^h \left(\sum_{i=2}^{n-1} 2^i v_i\right)^{t-h} = n^t - v_1^t.$$

It is easy to see that the left hand side is a polynomial in the variables $v_2, \ldots, v_n$ of total degree at most t with no free term. Thus we can write

$$\sum_{\substack{I=(I_2,\ldots,I_n)\in\mathbb{N}^{n-1}\\ 0<\sum I_i\leq t}} \delta_I v_s^{I_s} \cdots v_1^{I_1} = n^t - v_1^t,$$

where $\{\delta_I\}$ depend on n, t, s, as required. □

We are now ready to prove that a (d,t)-error reduction polynomial has a large L_1-norm.

Theorem 5.5 (L_1 bound)**.** *Let $n, d, t \in \mathbb{N}$ be such that $t < n$. Then, for every (d,t)-error reduction polynomial $\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, correlated or uncorrelated,*

$$L_1(\mathsf{Q}) \geq 2\left(\frac{n}{d}\right)^t - 1.$$

Proof. Let $(\gamma)_{m\in M(n,d)}$ be a (d,t)-coefficient vector. By Claim 5.2 it suffices to show that the L_1-norm of $(\gamma)_{m\in M(n,d)}$ is at least $2\left(\frac{n}{d}\right)^t-1$. To this end, we first recall that by Claim 5.4, for some $\{\delta_I\in\mathbb{R}\}_{I=(I_2,\ldots,I_n)\in\mathbb{N}^{n-1},0<\sum I_i\le t}$ we have that for every monomial $m=(v_1,\ldots,v_n)\in M(n,d)$ it holds that

$$\sum_{\substack{I=(I_2,\ldots,I_n)\in\mathbb{N}^{n-1}\\0<\sum I_i\le t}}\delta_I v_n^{I_n}\cdots v_2^{I_2}=n^t-v_1^t.$$

Dividing by $\frac{d^t}{2}$ gives

$$\sum_{\substack{I=(I_2,\ldots,I_n)\in\mathbb{N}^{n-1}\\0<\sum I_i\le t}}\frac{2\delta_I}{d^t}v_n^{I_n}\cdots v_2^{I_2}=2\left(\frac{n}{d}\right)^t-2\frac{v_1^t}{d^t}.$$

As $0\le v_1\le d$ (by the definition of $M(n,d)$), we see that

$$\sum_{\substack{I\in(I_2,\ldots,I_n)\in\mathbb{N}^{n-1}\\0<\sum I_i\le t}}\frac{2\delta_I}{d^t}v_n^{I_n}\cdots v_2^{I_2}\in\left[2\left(\frac{n}{d}\right)^t-2,2\left(\frac{n}{d}\right)^t\right].\tag{34}$$

Now, for every $I=(I_2,\ldots,I_n)\in\mathbb{N}^{n-1}$ such that $0<\sum I_i\le t$ we proceed by multiplying Equation (33) in Claim 5.3 by $\frac{2\delta_I}{d^t}$ and summing all equations together to get that

$$\sum_{\substack{I=(I_2,\ldots,I_n)\in\mathbb{N}^{n-1}\\0<\sum I_i\le t}}\sum_{(v_1,\ldots,v_n)\in M(n,d)}\frac{2\delta_I}{d^t}\gamma_{(v_1,\ldots,v_n)}\prod_{i=2}^n {v_i}^{I_i}=0.$$

Changing the order of summation,

$$\sum_{(v_1,\ldots,v_n)\in M(n,d)}\gamma_{(v_1,\ldots,v_n)}\sum_{\substack{I=(I_2,\ldots,I_n)\in\mathbb{N}^{n-1}\\0<\sum I_i\le t}}\frac{2\delta_I}{d^t}\prod_{i=2}^n {v_i}^{I_i}=0.\tag{35}$$

Next we multiply Equation (31) by $2\left(\frac{n}{d}\right)^t-1$ to get that

$$\sum_{m\in M(n,d)}\gamma_m\left(2\left(\frac{n}{d}\right)^t-1\right)=2\left(\frac{n}{d}\right)^t-1.$$

Subtracting Equation (35) from the above equation, we get that

$$\sum_{m\in M(n,d)}\gamma_m\left(2\left(\frac{n}{d}\right)^t-1-\sum_{\substack{I=(I_2,\ldots,I_n)\in\mathbb{N}^{n-1}\\0<\sum I_i\le t}}\frac{2\delta_I}{d^t}\prod_{i=2}^n {v_i}^{I_i}\right)=2\left(\frac{n}{d}\right)^t-1.$$

Hence,

$$\begin{aligned}
2\left(\frac{n}{d}\right)^t - 1 &= \sum_{m\in M(n,d)} \gamma_m \left(2\left(\frac{n}{d}\right)^t - 1 - \sum_{\substack{I=(I_2,\dots,I_n)\in\mathbb{N}^{n-1}\\ 0<\sum I_i\le t}} \frac{2\delta_I}{d^t}\prod_{i=2}^n {v_i}^{I_i}\right)\\
&\le \sum_{m\in M(n,d)} |\gamma_m| \left|2\left(\frac{n}{d}\right)^t - 1 - \sum_{\substack{I=(I_2,\dots,I_n)\in\mathbb{N}^{n-1}\\ 0<\sum I_i\le t}} \frac{2\delta_I}{d^t}\prod_{i=2}^n {v_i}^{I_i}\right|\\
&\le \sum_{m\in M(n,d)} |\gamma_m|\cdot 1,
\end{aligned}$$

where that the last inequality is by Equation (34). This gives us the required bound. □

6 L_0 Norm Bound, in the Non-Commutative Setting

In this section, we show that in the non-commutative setting, we can establish a bound on the L_0-norm of an error-reduction polynomial. Let us take the error reduction polynomial from Equation (9) as an example to illustrate the difference between the commutative and the non-commutative settings. In the commutative setting, Equation (9) takes the form

$$\mathsf{Q}(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3,\boldsymbol{x}_4) = 2\boldsymbol{x}_1^2\boldsymbol{x}_2 - \boldsymbol{x}_2^2.$$

However, in the non-commutative setting, to achieve $\varepsilon \ll \varepsilon_0$, the error reduction polynomial should be

$$\mathsf{Q}(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3,\boldsymbol{x}_4) = \boldsymbol{x}_1^2\boldsymbol{x}_2 + \boldsymbol{x}_2\boldsymbol{x}_1^2 - \boldsymbol{x}_2^2.$$

This shows that distinct terms in the non-commutative setting might be merged in the commutative setting. Thus, the L_0 norms in the commutative and non-commutative settings are quite different.

We first need some new definitions for the non-commutative setting.

Definition 6.1. Define

$$N(n) \triangleq \{(k, v_1, \dots, v_k)\colon k\ge 1,\ v_1,\dots,v_k\in[n], v_1+\cdots+v_k = n\}.$$

Then, any monomial over non-commutative variables $\boldsymbol{x}_1,\dots,\boldsymbol{x}_n$ such that the sum of the indices is n has form $\boldsymbol{x}_{v_1}\boldsymbol{x}_{v_2}\cdots\boldsymbol{x}_{v_k}$ for some $(k, v_1,\dots,v_k)\in N(n)$.

Let $\mathbb{R}_{\mathsf{nc}}[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ be the set of all real-coefficient degree-n homogeneous polynomials over non-commutative variables $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. Moreover, for

$$\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{(k,v_1,\ldots,v_k)\in N(n)} c_{k,v_1,\ldots,v_k} \boldsymbol{x}_{v_1} \ldots \boldsymbol{x}_{v_k},$$

recall that we define

$$L_0(\mathsf{Q}) = \sum_{(k,v_1,\ldots,v_k)\in N(n)} \mathbb{I}[c_{k,v_1,\ldots,v_k} \neq 0].$$

Definition 6.2. Define

$$N'(n) \triangleq \left\{ (k, v_1, \ldots, v_k, w_1, \ldots, w_{k+1}) \colon k \geq 0,\ v_i \in [n],\ w_i \in \mathbb{N},\ \sum_{i=1}^{k} v_i + \sum_{i=1}^{k+1} w_i = n \right\}.$$

For $n, d, t \in \mathbb{N}$ and a polynomial $\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in \mathbb{R}_{\mathsf{nc}}[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, we say Q is a (d,t)*-error reduction polynomial (for the non-commutative setting)*, if the total degree of Q is at most d, and there exist real coefficients $\{c_{\mathbf{v}}\}_{\mathbf{v}\in N'(n)}$ such that

$$\begin{aligned} &\mathsf{Q}(\boldsymbol{a}, \boldsymbol{a}^2 + \boldsymbol{\varepsilon}_2, \ldots, \boldsymbol{a}^n + \boldsymbol{\varepsilon}_n) - \boldsymbol{a}^n \\ &= \sum_{\substack{\mathbf{v}=(k,v_1,\ldots,v_k,w_1,\ldots,w_{k+1})\in N'(n) \\ k \geq t+1}} c_{\mathbf{v}} \boldsymbol{a}^{w_1} \boldsymbol{\varepsilon}_{v_1} \boldsymbol{a}^{w_2} \boldsymbol{\varepsilon}_{v_2} \cdots \boldsymbol{a}^{w_k} \boldsymbol{\varepsilon}_{v_k} \boldsymbol{a}^{w_{k+1}}. \end{aligned}$$

i.e., in $\mathsf{Q}(\boldsymbol{a}, \boldsymbol{a}^2 + \boldsymbol{\varepsilon}_2, \ldots, \boldsymbol{a}^n + \boldsymbol{\varepsilon}_n) - \boldsymbol{a}^n$, the total degree of $\boldsymbol{\varepsilon}_2, \ldots, \boldsymbol{\varepsilon}_n$ is at least $t+1$ in each non-zero term.

Then, we state and prove our bound for L_0 in the non-commutative setting. We also handle the correlated errors simultaneously.

Theorem 6.3. *Let $n > d' \geq d > t \geq 1$. If $\mathsf{Q}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in \mathbb{R}_{\mathsf{nc}}[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ is a (d', t)-error reduction polynomial (for the non-commutative setting), such that $\deg_{\boldsymbol{x}_1}(\mathsf{Q}) \leq d$, then*

$$L_0(\mathsf{Q}) \geq \frac{\binom{n}{t}}{\binom{d}{t}}.$$

Remark 6.4. Note that since the total degree of a polynomial is at least the degree of $\boldsymbol{x}_1$, Theorem 6.3 also implies that for any (d,t)-error reduction polynomial Q, we have $L_0(\mathsf{Q}) \geq \binom{n}{t}/\binom{d}{t}$.

We need to mention that the correlated error in the non-commutative setting is more complicated. In the commutative setting, for example, we have

$$(\boldsymbol{a}^{11} + \boldsymbol{\varepsilon}_{11}) = \boldsymbol{a}(\boldsymbol{a}^2 + \boldsymbol{\varepsilon}_2)(\boldsymbol{a}^8 + \boldsymbol{\varepsilon}_8).$$

However, for the non-commutative setting, the order of $\boldsymbol{a}$, $(\boldsymbol{a}^2+\boldsymbol{\varepsilon}_2)$, $(\boldsymbol{a}^8+\boldsymbol{\varepsilon}_8)$ can be arbitrary, and the order might even be different between the different appearances of $(\boldsymbol{a}^{11}+\boldsymbol{\varepsilon}_{11})$ in the expression of Q. Nonetheless, as long as the number of $\boldsymbol{a}$-s in the decomposition of each $(\boldsymbol{a}^r+\boldsymbol{\varepsilon}_r)$ is bounded by 1, degree-d error reduction polynomials (with correlated errors) can be captured by $\deg_{\boldsymbol{x}_1} \le d$. So Theorem 6.3 still gives a bound for L_0 in the correlated error setting.

Proof of Theorem 6.3. For $I \subseteq [n]$ define monomial $B_I(\boldsymbol{a},\boldsymbol{\varepsilon}) = \mathbf{y}_1\mathbf{y}_2\cdots\mathbf{y}_n$, here $\mathbf{y}_i = \boldsymbol{\varepsilon}$ if $i \in I$ and $\mathbf{y}_i = \boldsymbol{a}$ if $i \notin I$. Consider

$$\mathsf{R}(\boldsymbol{a},\boldsymbol{\varepsilon}) \triangleq \mathsf{Q}(\boldsymbol{a}+\boldsymbol{\varepsilon}, (\boldsymbol{a}+\boldsymbol{\varepsilon})^2 + (\boldsymbol{a}^2-(\boldsymbol{a}+\boldsymbol{\varepsilon})^2), \ldots, (\boldsymbol{a}+\boldsymbol{\varepsilon})^n + (\boldsymbol{a}^n - (\boldsymbol{a}+\boldsymbol{\varepsilon})^n)).$$

By the definition of (d,t)-error reduction polynomial, we can write

$$\begin{aligned}
&\mathsf{R}(\boldsymbol{a},\boldsymbol{\varepsilon}) - (\boldsymbol{a}+\boldsymbol{\varepsilon})^n \\
&= \sum_{\substack{\mathbf{v}=(k,v_1,\ldots,v_k,w_1,\ldots,w_{k+1})\in N'(n)\\ k\ge t+1}} c_{\mathbf{v}}(\boldsymbol{a}+\boldsymbol{\varepsilon})^{w_1}(\boldsymbol{a}^{v_1}-(\boldsymbol{a}+\boldsymbol{\varepsilon})^{v_1})\cdots(\boldsymbol{a}+\boldsymbol{\varepsilon})^{w_k}(\boldsymbol{a}^{v_k}-(\boldsymbol{a}+\boldsymbol{\varepsilon})^{v_k})(\boldsymbol{a}+\boldsymbol{\varepsilon})^{w_{k+1}} \\
&= (-1)^k \sum_{\substack{\mathbf{v}=(k,v_1,\ldots,v_k,w_1,\ldots,w_{k+1})\in N'(n)\\ k\ge t+1\\ \varnothing\ne I_1\subseteq[v_1],\ldots,\varnothing\ne I_k\subseteq[v_k]}} c_{\mathbf{v}}(\boldsymbol{a}+\boldsymbol{\varepsilon})^{w_1}B_{I_1}(\boldsymbol{a},\boldsymbol{\varepsilon})\cdots(\boldsymbol{a}+\boldsymbol{\varepsilon})^{w_k}B_{I_k}(\boldsymbol{a},\boldsymbol{\varepsilon})(\boldsymbol{a}+\boldsymbol{\varepsilon})^{w_{k+1}}.
\end{aligned}$$

The last equation can be written as

$$\sum_{I\subseteq[n],|I|\ge t+1} \lambda_I B_I(\boldsymbol{a},\boldsymbol{\varepsilon})$$

for some real numbers $\{\lambda_I\}_{I\subseteq[n]}$. Therefore,

$$\mathsf{R}(\boldsymbol{a},\boldsymbol{\varepsilon}) = \sum_{I\subseteq[n],|I|\le t} B_I(\boldsymbol{a},\boldsymbol{\varepsilon}) + \sum_{I\subseteq[n],|I|\ge t+1} (\lambda_I+1)B_I(\boldsymbol{a},\boldsymbol{\varepsilon}). \tag{36}$$

On the other hand, suppose

$$\mathsf{Q}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n) = \sum_{(k,v_1,\ldots,v_k)\in N(n)} \mu_{k,v_1,\ldots,v_k}\boldsymbol{x}_{v_1}\cdots\boldsymbol{x}_{v_k}.$$

Since $\deg_{\boldsymbol{x}_1}(\mathsf{Q}) \le d$, we know that for each non-zero $\mu_{k,v_1,\ldots,v_k}$, there are at most d 1-s in $v_1,\ldots,v_k$. Thus,

$$\mathsf{R}(\boldsymbol{a},\boldsymbol{\varepsilon}) = \sum_{\substack{(k,v_1,\ldots,v_k)\in N(n)\\ |\{i\in[k]\colon v_i=1\}|\le d}} \mu_{k,v_1,\ldots,v_k}\mathbf{y}_{v_1}\cdots\mathbf{y}_{v_k},$$

here $\mathbf{y}_1 \triangleq (\boldsymbol{a}+\boldsymbol{\varepsilon})$ and $\mathbf{y}_i \triangleq \boldsymbol{a}^i$ for $i \ge 2$. For each non-zero $\mu_{k,v_1,\ldots,v_k}$, $\mathbf{y}_{v_1}\cdots\mathbf{y}_{v_k}$ is a multiplication of some $(\boldsymbol{a}+\boldsymbol{\varepsilon})$-s and $\boldsymbol{a}$-s, and since $|\{i\in[k]\colon v_i=1\}| \le d$, we know there are at most d $(\boldsymbol{a}+\boldsymbol{\varepsilon})$-s in the multiplication. Thus, for each $\mathbf{v} = (k, v_1, \ldots, v_k) \in N(n)$ such that $|\{i \in [k] \colon v_i = 1\}| \leq d$, we can assign a set $J_{\mathbf{v}} \subseteq [n]$ such that $\mathbf{y}_{v_1} \cdots \mathbf{y}_{v_k} = B_{J_{\mathbf{v}}}(\boldsymbol{a}, \boldsymbol{a}+\boldsymbol{\varepsilon})$ and $|J_{\mathbf{v}}| \leq d$. Thus,

$$\begin{aligned} \mathsf{R}(\boldsymbol{a}, \boldsymbol{\varepsilon}) &= \sum_{\mathbf{v} \in N(n), \mu_{\mathbf{v}} \neq 0} \mu_{\mathbf{v}} B_{J_{\mathbf{v}}}(\boldsymbol{a}, \boldsymbol{a}+\boldsymbol{\varepsilon}) \\ &= \sum_{\substack{\mathbf{v} \in N(n), \mu_{\mathbf{v}} \neq 0 \\ I \subseteq J_{\mathbf{v}}}} \mu_{\mathbf{v}} B_I(\boldsymbol{a}, \boldsymbol{\varepsilon}). \end{aligned} \tag{37}$$

Let us compare Equation (36) and Equation (37). Consider writing $\mathsf{R}(\boldsymbol{a}, \boldsymbol{\varepsilon})$ into a weighted sum of $B_I(\boldsymbol{a}, \boldsymbol{\varepsilon})$-s $(I \subseteq [n])$. By Equation (37) we know that the number of non-zero $B_I(\boldsymbol{a}, \boldsymbol{\varepsilon})$-s $(|I| = t)$ is bounded above by

$$\sum_{\mathbf{v} \in N(n), \mu_{\mathbf{v}} \neq 0} \binom{|J_{\mathbf{v}}|}{t} \leq L_0(\mathsf{Q}) \cdot \binom{d}{t}.$$

On the other hand by Equation (36) we know the coefficient of each $B_I(\boldsymbol{a}, \boldsymbol{\varepsilon})$ $(|I| = t)$ is non-zero, so the number of non-zero $B_I(\boldsymbol{a}, \boldsymbol{\varepsilon})$-s $(|I| = t)$ is $= \binom{n}{t}$. Note that this crucially depends on the non-commutative setting. Therefore $\binom{n}{t} \leq L_0(\mathsf{Q}) \cdot \binom{d}{t}$, which completes the proof. □

7 A Simultaneous L_1 and Degree Bound

Recall our discussion from Section 2.3 about the WPRGs error reduction framework, wherein we use an auxiliary PRG G_{aux} to "derandomize" each monomial. As we also noted in Section 2.3, a very small degree d may remove the need to use G_{aux}, and thereby avoid the dependence on $L_1(\mathsf{Q})$. As previous results show, unfortunately, d must be large. A contrarian reader may suggest the following avenue towards evading the degree barrier: Suppose that Q, *restricted* to high-degree monomials, had small L_1 norm. Then, we could derandomize high-degree monomials and use independent seeds to approximate the low-degree monomials. This way, we could conceivably avoid paying for the (large) L_1 norm of the polynomial itself. In this section, we show that this cannot be the case. For concreteness, let us focus on (a specific instantiation of) correlated errors[11], and readily start with the reduced setting (which we know, by Claim 3.4 suffices).

[11] Recall that in an error reduction polynomial for correlated errors, as used in the "WPRG step" of Section 2.3, the total degree is counted in terms of the number of approximations of A^{2^i} (including multiplicities). This is a suitable notion of degree when we need to pay for each power-of-two that we approximate, like in Nisan's PRG. In Theorem 7.1 we consider individual degrees, which in particular bound the total degree, as defined here.

Theorem 7.1 (simultaneous bound)**.** *Let $n, d \in \mathbb{N}$ such that $d < n$ and set $s = \log n$, where we assume that n is a power of 2. Let $\mathsf{Q}(\boldsymbol{x}_0, \dots, \boldsymbol{x}_s)$ be a reduced $(d,1)$-error reduction polynomial for correlated errors. Further, assume that for any ε_0 and $a \in [0,1]$, and any $\widetilde{a^2}, \widetilde{a^4}, \dots, \widetilde{a^{2^s}}$ such that $\left| a^{2^i} - \widetilde{a^{2^i}} \right| \le \varepsilon_0$, it holds that*

$$\left| \mathsf{Q}\left(a, \widetilde{a^2}, \widetilde{a^4}, \dots, \widetilde{a^{2^s}}\right) - a^n \right| \le \tau$$

for some $0 < \tau = \tau(n, \varepsilon_0) \le \frac{\varepsilon_0}{4}$.[12] *Then, $L_1(\mathsf{Q}_R) = \Omega(n/d)$, where Q_R is the restriction of Q to monomials of total degree at least $\frac{\log n}{2 \log d}$.*

Proof. Write $\mathsf{Q} = \sum_{\mathbf{I} \in \mathcal{I}} c_{\mathbf{I}} \boldsymbol{x}^{\mathbf{I}}$ where $\mathbf{I} \subseteq \{0, \dots, d\}^{s+1}$, and recall that we can assume that for every $\mathbf{I} \in \mathcal{I}$ for $\mathbf{I} = (i_0, \dots, i_s)$ has effective degree n, namely $w(\mathbf{I}) = \sum_{k=0}^{s} 2^k \cdot i_k = n$. We first observe that any monomial in Q which has $\boldsymbol{x}$ a term (i.e., $i_0 \ge 1$) must have high degree.

Claim 7.2. *For every $\mathbf{I} = (i_0, \dots, i_s) \in \mathcal{I}$ in which $i_0 \ge 1$ it holds that $|\mathbf{I}| \ge \frac{\log n}{2 \log d}$.*

Proof. Since $w(\mathbf{I}) = n$, we can write $\sum_{k=1}^{s} 2^k \cdot i_k = n - i_0$. This readily implies that there must be at least $\frac{\log n}{2 \log d}$ values of k for which $i_k \ne 0$. □

Now, write Q as $\mathsf{Q}(\boldsymbol{x}_0, \dots, \boldsymbol{x}_s) = \mathsf{Z}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_s) + \mathsf{R}(\boldsymbol{x}_0, \dots, \boldsymbol{x}_s)$, where Z contains all $\mathbf{I} = (i_0, \dots, i_s) \in \mathcal{I}$ for which $i_0 = 0$, and similarly R contains all $\mathbf{I}$-s for which $i_0 \ne 0$. Set $a_1 = 1$, and $a_2 = 1 - \frac{\varepsilon_0}{n}$. We can write

$$\mathsf{Q}(a_1, 1, \dots, 1) = \mathsf{Z}(1, \dots, 1) + \mathsf{R}_1(a_1, 1, \dots, 1)$$

and

$$\mathsf{Q}(a_2, 1, \dots, 1) = \mathsf{Z}(1, \dots, 1) + \mathsf{R}_2(a_2, 1, \dots, 1),$$

observing that indeed, the "Z part" in both polynomials is the same. Moreover, each $\mathbf{I}$ in R satisfies $|\mathbf{I}| \ge \frac{\log n}{2 \log d}$, by Claim 7.2. Now, since $|a_2^n - 1| \le 1 - (1 - \varepsilon_0/n)^n \le \varepsilon_0$ (and obviously the same is true for a_1), we have that

$$\tau \ge |a_1^n - \mathsf{Q}(a_1, 1, \dots, 1)| = |1 - \mathsf{Z}(1, \dots, 1) - \mathsf{R}_1(a_1, 1, \dots, 1)|\,, \tag{38}$$

and

$$\tau \ge |a_2^n - \mathsf{Q}(a_2, 1, \dots, 1)| = |a_2^n - \mathsf{Z}(1, \dots, 1) - \mathsf{R}_2(a_2, 1, \dots, 1)|\,, \tag{39}$$

[12] Note that we do not simply require the linear terms in $\boldsymbol{\varepsilon}$ to vanish – which would naively say that $\tau \le L_1(\mathsf{Q})\varepsilon_0^2$. Indeed, we *must* assume that establishing the error-reduction capabilities of Q do not go through the L_1 norm, since we already *know* it's large.

recalling that $\tau = \tau(n, \varepsilon_0)$. Writing $\rho = 1 - \mathsf{Z}(1, \ldots, 1)$, Equation (38) tells us that $|\mathsf{R}_1(a_1, 1, \ldots, 1) - \rho| \leq \tau$, and Equation (39) tells us that $|\mathsf{R}_2(a_2, 1, \ldots, 1) - \rho| \geq (1 - a_2^n) - \tau$. From these inequalities we can infer that

$$|\mathsf{Q}(a_2, 1, \ldots, 1) - \mathsf{Q}(a_1, 1, \ldots, 1)| = |(\mathsf{R}_1(a_1, 1, \ldots, 1) - \rho) - (\mathsf{R}_2(a_2, 1, \ldots, 1) - \rho)| \geq \\ |\mathsf{R}_2(a_2, 1, \ldots, 1) - \rho| - |\mathsf{R}_1(a_1, 1, \ldots, 1) - \rho| \geq (1 - a_2^n) - 2\tau \geq \varepsilon_0/2 - 2\tau,$$

where the last inequality follows from the fact that $a_2^n \leq e^{-\varepsilon_0} \leq 1 - \frac{\varepsilon_0}{2}$. Since $\tau \leq \varepsilon_0/4$, we get that $|\mathsf{Q}(a_2, 1, \ldots, 1) - \mathsf{Q}(a_1, 1, \ldots, 1)| \geq \frac{\varepsilon_0}{4}$. But

$$|\mathsf{Q}(a_2, 1, \ldots, 1) - \mathsf{Q}(a_1, 1, \ldots, 1)| = \left| \sum_{\mathbf{I} \in \mathcal{I} : i_0 \neq 0} c_{\mathbf{I}} \left(a_2^{i_1} - 1\right) \right| \leq \sum_{\mathbf{I} \in \mathcal{I} : i_0 \neq 0} |c_{\mathbf{I}}| \cdot \left|a_2^{i_1} - 1\right|.$$

Recall that $i_1 \leq d$, so $|a_2^{i_1} - 1| \leq \frac{\varepsilon_0 d}{n}$. Thus,

$$\sum_{\mathbf{I} \in \mathcal{I} : i_0 \neq 0} |c_{\mathbf{I}}| \geq \frac{n}{4d},$$

as desired. □

References

[AKL+79] Romas Aleliunas, Richard M. Karp, Richard J. Lipton, László Lovász, and Charles Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. In *20th Annual Symposium on Foundations of Computer Science (FOCS 1979)*, pages 218–223. IEEE, 1979.

[AKM+20] AmirMahdi Ahmadinejad, Jonathan Kelner, Jack Murtagh, John Peebles, Aaron Sidford, and Salil Vadhan. High-precision estimation of random walks in small space. In *61st Annual Symposium on Foundations of Computer Science (FOCS 2020)*, pages 1295–1306. IEEE, 2020.

[AL96] Noga Alon and Michael Luby. A linear time erasure-resilient code with nearly optimal recovery. *IEEE Transactions on Information Theory*, 42(6):1732–1736, 1996.

[BCG18] Mark Braverman, Gil Cohen, and Sumegha Garg. Hitting sets with near-optimal error for read-once branching programs. In *50th Annual Symposium on Theory of Computing (STOC 2018)*, pages 353–362. ACM, 2018.

[BCP83] Allan Borodin, Stephen Cook, and Nicholas Pippenger. Parallel computation for well-endowed rings and space-bounded probabilistic machines. *Information and Control*, 58(1-3):113–136, 1983.

[BRRY14] Mark Braverman, Anup Rao, Ran Raz, and Amir Yehudayoff. Pseudorandom generators for regular branching programs. *SIAM Journal on Computing*, 43(3):973–986, 2014.

[CCMP23] Gil Cohen, Itay Cohen, Gal Maor, and Yuval Peled. Derandomized squaring: An analytical insight into its true behavior. In *Electronic Colloquium on Computational Complexity (ECCC)*, number 183, 2023. Manuscript.

[CDR$^+$21] Gil Cohen, Dean Doron, Oren Renard, Ori Sberlo, and Amnon Ta-Shma. Error reduction for weighted PRGs against read once branching programs. In *36th Computational Complexity Conference (CCC 2021)*, pages 22:1–22:17. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[CHL$^+$23] Lijie Chen, William M. Hoza, Xin Lyu, Avishay Tal, and Hongxun Wu. Weighted pseudorandom generators via inverse analysis of random walks and shortcutting. In *64th Annual Symposium on Foundations of Computer Science (FOCS 2023)*, pages 1224–1239. IEEE, 2023.

[CL20] Eshan Chattopadhyay and Jyun-Jie Liao. Optimal error pseudodistributions for read-once branching programs. In *35th Computational Complexity Conference (CCC 2020)*, pages 25:1–25:27. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[CY21] Gil Cohen and Tal Yankovitz. Rate amplification and query-efficient distance amplification for linear LCC and LDC. In *36th Computational Complexity Conference (CCC 2020)*, volume 200, pages 1:1–1:57. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[CY22] Gil Cohen and Tal Yankovitz. LCC and LDC: tailor-made distance amplification and a refined separation. In *49th EATCS International Conference on Automata, Languages, and Programming (ICALP 2022)*, volume 229, pages 44:1–44:20. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[De11] Anindya De. Pseudorandomness for permutation and regular branching programs. In *26th Conference on Computational Complexity (CCC 2011)*, pages 221–231. IEEE, 2011.

[Din07] Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM (JACM)*, 54(3):12–es, 2007.

[DPT24] Dean Doron, Edward Pyne, and Roei Tell. Opening up the distinguisher: A hardness to randomness approach for **BPL** = **L** that uses properties of **BPL**. In *56th Annual Symposium on Theory of Computing (STOC 2024, forthcoming)*. ACM, 2024.

[dST$^+$21] Alexandre d'Aspremont, Damien Scieur, Adrien Taylor, et al. Acceleration methods. *Foundations and Trends® in Optimization*, 5(1-2):1–245, 2021.

[DT23] Dean Doron and Roei Tell. Derandomization with minimal memory footprint. In *38th Computational Complexity Conference (CCC 2023)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2023.

[Hoz21] William M. Hoza. Better pseudodistributions and derandomization for space-bounded computation. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[Hoz22] William M. Hoza. Recent progress on derandomizing space-bounded computation. *Bulletin of the EATCS*, (138):114–143, 2022.

[HPV21] William M. Hoza, Edward Pyne, and Salil Vadhan. Pseudorandom generators for unbounded-width permutation branching programs. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[HZ20] William M. Hoza and David Zuckerman. Simple optimal hitting sets for small-success **RL**. *SIAM Journal on Computing*, 49(4):811–820, 2020.

[INW94] Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In *26th Annual Symposium on Theory of Computing (STOC 1994)*, pages 356–364. ACM, 1994.

[Jun81] H. Jung. Relationships between probabilistic and deterministic tape complexity. In *International Symposium on Mathematical Foundations of Computer Science (MFCS 1981)*, pages 339–346. Springer, 1981.

[KMRZS17] Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. High-rate locally correctable and locally testable codes with sub-polynomial query complexity. *Journal of the ACM (JACM)*, 64(2):1–42, 2017.

[KNP11] Michal Kouckỳ, Prajakta Nimbhorkar, and Pavel Pudlák. Pseudorandom generators for group products. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 263–272, 2011.

[Koz20] Daniel Kozlov. Simplifying the BCG PRPD for space bounded computation, 2020. Available at https://www.cs.tau.ac.il/~amnon/Students/daniel.kozlov.pdf.

[KvM02] Adam R. Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM Journal on Computing*, 31(5):1501–1526, 2002.

[MRSV17] Jack Murtagh, Omer Reingold, Aaron Sidford, and Salil Vadhan. Derandomization beyond connectivity: Undirected laplacian systems in nearly logarithmic space. In *58th Annual Symposium on the Foundations of Computer Science (FOCS 2017)*, pages 801–812. IEEE, 2017.

[MRSV19] Jack Murtagh, Omer Reingold, Aaron Sidford, and Salil Vadhan. Deterministic approximation of random walks in small space. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*, pages 42:1–42:22. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2019.

[Nis92] Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

[PV21] Edward Pyne and Salil Vadhan. Pseudodistributions that beat all pseudorandom generators. In *36th Computational Complexity Conference (CCC 2021)*, pages 33:1–33:15. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[Rei08] Omer Reingold. Undirected connectivity in log-space. *Journal of the ACM (JACM)*, 55(4):1–24, 2008.

[RRV99] Ran Raz, Omer Reingold, and Salil Vadhan. Error reduction for extractors. In *40th Annual Symposium on the Foundations of Computer Science (FOCS 1999)*, pages 191–201. IEEE, 1999.

[RV05] Eyal Rozenman and Salil Vadhan. Derandomized squaring of graphs. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2005)*, pages 436–447. Springer, 2005.

[ST04] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *36th*

Anual Symposium on Theory of Computing (STOC 2004), pages 81–90. ACM, 2004.

[Ste12] Thomas Steinke. Pseudorandomness for permutation branching programs without the group theory. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 19, 2012. Manuscript.

[Ta-17] Amnon Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In *49th Annual Symposium on Theory of Computing (STOC 2017)*, pages 238–251. ACM, 2017.