Electronic Colloquium on Computational Complexity, Report No. 179 (2025)

Wide Replacement Products Meet Gray Codes: Toward Optimal Small-Bias Sets

Gil Cohen[*] Itay Cohen[†]

November 12, 2025

Abstract

Optimal small-bias sets sit at the crossroads of coding theory and pseudorandomness. Reaching optimal parameters would, in particular, meet the long-standing goal of matching the Gilbert–Varshamov bound for binary codes in the high-distance regime. In a breakthrough, Ta-Shma [TS17] constructed near-optimal small-bias sets via the Rozenman–Wigderson expander-walk framework, using the wide–replacement product to maintain s "secure" registers and to route the walk through them. Within this framework, two barriers remain en route to optimal small-bias sets: (i) the cost of maintaining registers and (ii) limitations inherited from spectral-gap bounds for expanders.

We overcome the first—arguably the more critical—barrier. Our key technical insight is that registers can be *reused* even after they are exposed. Using a Gray–code–style reuse schedule, we recycle the same s registers exponentially many times in s, thereby reducing the register–maintenance cost exponentially. This yields the first improvement over Ta-Shma's construction in nearly a decade—quantitatively modest but an important first step toward a truly optimal construction. The remaining barrier is fairly standard in isolation; the challenge is to overcome it in concert with our register-reuse framework.

[*]Tel Aviv University. gil@tauex.tau.ac.il. Supported by ERC starting grant 949499 and by the Israel Science Foundation grant 2989/24.

[†]Tel Aviv University. itay3@mail.tau.ac.il. Supported by ERC starting grant 949499.

ISSN 1433-8092

Contents

1 Introduction

Small-bias sets sit at the crossroads of pseudorandomness and coding theory. From the pseudorandomness viewpoint, a small-bias set can be thought of as a pseudorandom generator against linear (parity) tests. Specifically, a set $S \subseteq \{0,1\}^n$ is called *ε-biased* if for every nonzero $\tau \in \{0,1\}^n$,

$$\left|\mathbb{E}_{s\in S}\left[(-1)^{\langle s,\tau\rangle}\right]\right| \le \varepsilon.$$

For any nonzero τ, a uniformly random $u \in \{0,1\}^n$ satisfies $\mathbb{E}\left[(-1)^{\langle u,\tau\rangle}\right] = 0$. Thus an ε-biased set behaves like the uniform distribution with respect to parity tests and can be viewed as the output distribution of a PRG that fools such tests within error ε.

The main challenge is to explicitly construct, for given n and ε, an ε-biased set $S \subseteq \{0,1\}^n$ whose size is as small as possible. A standard probabilistic argument shows that ε-biased sets of size $O(n/\varepsilon^2)$ exist. This bound is essentially optimal: up to a multiplicative $\log(1/\varepsilon)$ factor, it matches the lower bound given by the classical linear-programming bound [MRRW77].

From the coding-theoretic perspective, an ε-biased set S yields a binary linear code of block length $|S|$, relative distance $\delta = \frac{1}{2} - \varepsilon$, and rate $\rho = \frac{n}{|S|}$. In particular, a construction matching the probabilistic bound would give an explicit code meeting the Gilbert–Varshamov bound [Gil52, Var57]—a decades-old open problem in coding theory. In pseudorandomness, small-bias sets are key ingredients in PRG constructions that go beyond linear tests [Vio09, FK18, CHHL19] and in the construction of randomness extractors [Raz05, CRS12]; they also have applications across adjacent areas, including cryptography, combinatorics, and quantum protocols.

Given their importance, small-bias sets have been studied extensively. Naor and Naor gave a construction with linear (optimal) dependence on n but suboptimal dependence on ε, namely $O(n/\varepsilon^c)$ for some constant $c > 2$ [NN93]. Alon, Goldreich, Håstad, and Peralta provided several constructions with optimal dependence on ε but suboptimal in n, particularly $O(n^2/\varepsilon^2)$ [AGHP93]. From the coding viewpoint, keeping the n-dependence linear is crucial for the rate, while from the pseudorandomness perspective polynomial dependence on n is often acceptable, so the ε-dependence takes center stage. Several other constructions are known. For example, concatenating optimal algebraic–geometric codes with the Hadamard code yields a set of size $O(n/\varepsilon^3)$; using Hermitian codes instead gives an incomparable bound of $O\big((n/\varepsilon^2)^{5/4}\big)$ [BATS13].

About a decade ago, Ta-Shma [TS17] achieved a breakthrough with an explicit construction of size very close to optimal:

$$\frac{n}{\varepsilon^{2+\alpha}} \qquad \text{where} \qquad \alpha = \widetilde{O}\left(\left(\frac{1}{\log(1/\varepsilon)}\right)^{1/3}\right). \text{[1]} \tag{1}$$

1.1 Bias reduction

Ta-Shma's construction is based on *bias reduction*: starting from an ε_0-biased set (where ε_0 can even be a small constant), it transforms it into an ε-biased set for the target parameter ε. Since linear-in-n starting constructions are available, the heart of the matter is how the bias evolves throughout the reduction process.

A simple and natural way to reduce bias is this: given an ε_0-biased set S, consider

$$S + S \triangleq \{\, s_1 \oplus s_2 : s_1, s_2 \in S \,\},$$

where $\oplus$ denotes bitwise XOR. In pseudorandomness terms, the new sample space is obtained by drawing two independent uniform elements from S and outputting their XOR. This approach is, however, expensive: treating $S+S$ as a multiset [2], its size is $|S+S| = |S|^2$, so in particular the construction incurs at least quadratic dependence on n.

In unpublished work, Rozenman and Wigderson (credited in Bogdanov's 2012 complexity course notes) proposed derandomizing this idea using expanders. Instead of sampling two independent $s_1, s_2 \in S$, they suggested taking an expander whose vertex set is S, sample a random edge $\{s_1, s_2\}$, and output $s_1 \oplus s_2$. They showed that the resulting bias is at most $\varepsilon_0^2 + \omega$, where ω is the spectral expansion of the expander - the penalty one pays for the derandomization. Iterating this transformation from a constant-bias starting point yields an ε-biased set of size about n/ε^4.

An alternative to repeatedly applying the derandomized "sum-of-two" step is to take a longer walk on the expander G. In particular, for a parameter $t \geq 2$, take a length-t random walk on G and output the bitwise XOR of the t vertices visited. This yields essentially the same sample-space size as the iterated edge-based version.

While the random walk approach did not beat the best known constructions at the time, it was the starting point for Ta-Shma's construction. He observed that, in the random-walk approach, two distinct effects account for the exponent 4 in the sample-space size n/ε^4. The first is the essentially unavoidable *quadratic* relation between the degree d of a d-regular expander and its spectral expansion ω; even for Ramanujan graphs one has $\omega \approx \frac{2}{\sqrt{d}}$. The second factor-of-two loss is the part Ta-Shma managed to eliminate almost entirely.

[1] Throughout, we use the convention that $\widetilde{O}(\cdot)$ suppresses polylogarithmic factors in its argument; in our bounds it hides a factor of $(\log\log(1/\varepsilon))^{O(1)}$.

[2] Despite the name, small-bias "sets" are typically allowed to be multisets. This is natural for two reasons: (i) basic operations on small-bias sets (e.g., $S + S$) can introduce duplicates, and (ii) from the pseudorandomness viewpoint we often treat S as inducing a (possibly nonuniform) distribution over $\{0,1\}^n$.

1.2 Neutralizing adversarial interference

The second factor-of-two arises from the adversary's ability to correlate the walk with a fixed linear test τ. A length-t walk on G is governed by the operator W^t, where W is the random-walk (normalized adjacency) matrix of G. In the presence of a linear test, an additional operator is interleaved. The test τ induces a $\{\pm 1\}$ labeling of the vertices: for each $s \in S$, $\chi_\tau(s) = (-1)^{\langle \tau, s \rangle}$. Let Π be the diagonal matrix with $(\Pi)_{s,s} = \chi_\tau(s)$. Writing $\mathbf{1}$ for the normalized all-ones vector, the bias against τ of the new sample space is

$$\left| \mathbf{1}^\top \, \Pi \, (W\Pi)^{t-1} \, \mathbf{1} \right|.$$

The interleaving of Π with W captures the "adversarial interference" responsible for the extra factor-of-two loss. Informally, the sign flips introduced by Π can neutralize every other step of the walk, effectively wasting half the steps.

The key insight underlying Ta-Shma's improvement is to restrict, at each step, the set of neighbors a vertex of G may transition to in a manner that is "hidden" from the adversary. In this way, the effect of Π on the walk remains largely under our control. Ta-Shma implements this approach using the *s-wide replacement product*, introduced by Ben-Aroya and Ta-Shma [BATS11] for the purpose of constructing near-Ramanujan graphs.

Informally, the wide–replacement product uses a small "gadget" graph H to subsample the next vertex in a random walk. It extends operators such as the zig–zag product [RVW00] and derandomized squaring [RV05], and has the advantage of saving randomness: subsampling neighbors uses fewer random bits than choosing a uniformly random neighbor. When H is itself an expander, this subsampling preserves mixing well, so the walk remains well behaved while using much less randomness.

Ta-Shma's key observation is that, beyond its randomness-saving role—which is also crucial for keeping the sample space small in our setting—the wide–replacement product can be used to *hide* information about steps of the random walk, provided it is implemented appropriately. To this end, we will require the gadget H to have a specific structure: think of H as consisting of s *registers* $R_1, \ldots, R_s$, and at step i we use register R_i. This can be achieved by taking H to be the Cayley graph of a group that is a direct product of s groups. Ta-Shma showed that a length-s walk can be implemented using these s registers, and the net effect is that, instead of losing one in every two steps, we effectively lose only one out of every s steps.

Using more registers reduces the bias, but it also increases the sample-space size—exponentially in s. Balancing these two effects (by choosing s optimally) yields an ε-biased set whose size is as in Equation (1).

1.3 Our results

Ta-Shma's construction is already nearly optimal, but the importance of the problem calls for a truly optimal solution. This is especially compelling from the coding-theoretic viewpoint: matching the Gilbert–Varshamov bound would be a major breakthrough. From the pseudorandomness perspective, linear tests are among the simplest classes, so constructing an optimal PRG against them is a particularly natural goal. Indeed, to the best of our knowledge, no natural test class currently admits a PRG with optimal parameters; linear tests may be the most compelling candidate.

The goal of this paper is to advance this research program. While we do achieve a modest quantitative improvement over Ta-Shma's result, we view this as secondary. Our main technical contribution is to eliminate almost entirely one of the two barriers identified by Ta-Shma to further progress in his framework. We believe this is the harder and more substantial of the two, and we focus our efforts on overcoming it.

This barrier is related to the cost of maintaining s registers. As noted, it inflates the sample–space size by a factor exponential in s, which we then balance against the bias, whose magnitude decreases with the walk length. To obtain a better construction within this framework, one must extract more "juice" from the same register budget—namely, enable substantially longer walks without increasing s.

Our key insight is that, if done carefully, registers can be *reused*. Once a register has been used it becomes "compromised"—an adversary may correlated with it—so it no longer hides the walk by itself. Nevertheless, with a carefully designed reuse schedule, we show that s registers suffice to perform *exponentially* long walks—in particular, of length $2^{s/2}$. This dramatically improves the tradeoff in that aspect.

Using this idea, we obtain a modest improvement over Ta-Shma's construction of small-bias sets, even without addressing the second barrier.

Theorem 1.1. *For every $n, \varepsilon > 0$ there exists an explicit ε-biased set $S \subseteq \{0,1\}^n$ of size*

$$\frac{n}{\varepsilon^{2+\alpha}} \qquad \textit{where} \qquad \alpha = \widetilde{O}\left(\left(\frac{1}{\log(1/\varepsilon)}\right)^{1/2}\right).$$

The register reuse pattern underlying our construction is a variant of the flip–index sequence of the binary-reflected Gray code on s coordinates. These Gray-style sequences are defined recursively and yield patterns such as 1 2 1 for two registers and 1 2 1 3 1 2 1 for three registers, and so on. We show that, in the absence of the interleaved operator Π, these sequences enable long walks on the wide–replacement product in the standard expander setting. Although the unmodified Gray pattern does not suffice in the context of bias reduction, once Π is present, we introduce a tailored variant that does.

1.3.1 Beyond the current barriers for small-bias sets

The second barrier in Ta-Shma's framework is the inherent factor-two loss in the degree/expansion tradeoff, $\omega \approx \frac{2}{\sqrt{d}}$. To push further, we must bypass this bottleneck. In our setting, the gadget graph required for the bias–reduction step has additional structure that in fact precludes Ramanujan optimality, so instead of the constant 2 we suffer a super-constant loss. At any rate, we would like to avoid this factor altogether.

This may be attainable by switching to a non-backtracking variant of our procedure—or, essentially equivalently, by using a directed gadget. The main challenge is to integrate such a modification with the machinery developed here. An alternative is to employ *rotating expanders* [CM23], which keep us within the undirected realm. Although each individual expander carries a factor-2 (or higher) penalty, suitable compositions avoid exponential blow-up and incur only a linear loss. Here too, the main challenge is to integrate our framework with rotating expanders; we leave this to future work.

1.4 Related work

Ta-Shma's construction [TS17] has received significant attention over the past decade. Here we briefly review the works most relevant to ours.

Blanc and Doron [BD22] proposed an approach to improved small-bias sets based on hypergraphs. They showed that a result quantitatively similar to Theorem 1.1 follows if one can construct a certain hypergraph with suitable parameters. Although a random hypergraph has the required parameters, no explicit construction is known. Hypergraphs with better parameters—if they exist—could yield improved constructions.

As for decoding, no decoding algorithm was provided in [TS17]. This was later remedied by Jeronimo, Quintana, Srivastava, and Tulsiani [JQST20, JST21], who showed that a slight variant of Ta-Shma's codes is efficiently decodable—indeed, in almost linear time in n. Blanc and Doron [BD22] obtained an improved construction with a better rate. We leave the decodability of our codes to future work.

An interesting extension of [TS17] was obtained by Jeronimo, Mittal, Roy, and Wigderson, who generalized Ta-Shma's bias-amplification analysis from scalars to matrices of arbitrary dimension [JMRW25]. We leave extending our construction to this matrix-valued setting to future work.

1.5 Organization

The wide–replacement product—and especially its use for bias reduction—is technically involved. Nevertheless, there is a clean, intuitive picture, and we begin by developing it before adding the machinery our results require. In Section 2 we give a detailed exposition of Ta-Shma's bias-reduction framework. Along the way we preview several ideas used later by our improved construction and develop a diagrammatic viewpoint that will guide the analysis. Next, Section 3 offers an informal overview of our construction. Readers who prefer to jump straight to the formal sections may safely skip these two sections. Section 4 formally develops the framework for working with the wide–replacement product. Section 5 introduces the random-walk patterns we employ, and Section 6 analyzes their interaction with the interleaved operators. Finally, Section 7 assembles the pieces and proves Theorem 1.1.

2 An Exposition to Ta-Shma's Construction

2.1 Rozenman–Wigderson bias reduction

As discussed in Section 1, Rozenman–Wigderson use expanders to derandomize the $S + S$ construction, where S is an ε_0-biased set. Let $G = (V, E)$ be a d-regular graph on n vertices. Let W denote the normalized adjacency (random-walk) matrix of G, with eigenvalues $1 = \omega_1 \geq \omega_2 \geq \cdots \geq \omega_n$. We say that G has ω-*spectral expansion*, where $\omega \triangleq \max\{|\omega_2|, |\omega_n|\}$. The key property for our purposes is that W fixes the normalized all-ones vector $\mathbf{1}$ (since it is an eigenvector with eigenvalue $\omega_1 = 1$) and contracts every vector in its orthogonal complement $\mathbf{1}^\perp$ by at least a factor ω. That is, for every $v \in \mathbf{1}^\perp$, $\|Wv\| \leq \omega \|v\|$. Moreover, $\mathbf{1}^\perp$ is an invariant subspace of W: $Wv \in \mathbf{1}^\perp$ for all $v \in \mathbf{1}^\perp$.

The Rozenman–Wigderson bias–reduction idea proceeds as follows. Let $S \subseteq \{0,1\}^n$ be an ε_0-biased set, and let $G = (S, E)$ be a d-regular graph on the vertex set S with ω-spectral expansion. That is, we identify elements of S with the vertices of G. In this setup, the naïve reduction considered above takes all pairs of vertices, i.e., sample two independent uniform vertices of G and take their bitwise XOR. Rozenman–Wigderson's idea is to *derandomize* this step by restricting to the edge set, namely, the small-bias set is given by

$$S +_G S \triangleq \{\, s_u + s_v \;:\; uv \in E \,\}.$$

This new set has size

$$m \triangleq |S +_G S| = |E| = \frac{nd}{2}.$$

As a warm-up, let us analyze this construction and prove

Lemma 2.1. *$S +_G S$ is an ε-biased set for $\varepsilon = \varepsilon_0^2 + \omega$.*

Proof. Fix a nonzero $\tau \in \{0,1\}^n$. For each $s \in S$, label the vertex s by $(-1)^{\langle \tau, s\rangle}$. To evaluate the bias, we need to compare the fraction of edges in G whose endpoints have the same label (either $++$ or $--$) with the fraction whose endpoints have different labels ($+-$ or $-+$). To this end, define $\pi : S \to \mathbb{R}$ by $\pi(s) = \frac{1}{\sqrt{|S|}}(-1)^{\langle \tau, s\rangle}$ and view π as a column vector indexed by S. The bias of the new set is

$$\left|\mathbb{E}_{uv\sim E}[\pi(u)\pi(v)]\right| \;=\; \left|\pi^\top W\pi\right|,$$

where the expectation is over a uniformly random edge of G.

Recall that by our convention, $\mathbf{1} \in \mathbb{R}^{|S|}$ denotes the normalized all-ones vector (so $\|\mathbf{1}\| = 1$). Then the (signed) bias of the test τ is

$$\varepsilon_\tau \;\triangleq\; \mathbb{E}_{s\in S}\left[(-1)^{\langle\tau,s\rangle}\right] \;=\; \langle\pi,\mathbf{1}\rangle,$$

which we know satisfies $|\varepsilon_\tau| \leq \varepsilon_0$. Accordingly, we decompose

$$\pi \;=\; \varepsilon_\tau\,\mathbf{1} \;+\; \pi^\perp, \qquad \text{with } \;\pi^\perp \in \mathbf{1}^\perp.$$

As $\mathbf{1}$ and $\mathbf{1}^\perp$ are invariant spaces of W, and since $W\mathbf{1} = \mathbf{1}$, we have that

$$\begin{aligned}\pi^\top W\pi &= \left(\varepsilon_\tau\mathbf{1} + \pi^\perp\right)^\top W\left(\varepsilon_\tau\mathbf{1} + \pi^\perp\right)\\ &= \varepsilon_\tau^2 + \left(\pi^\perp\right)^\top W\pi^\perp.\end{aligned}$$

Now, by Cauchy-Schwartz inequality, and since $\|\pi^\perp\| \leq 1$,

$$\left|\left(\pi^\perp\right)^\top W\pi^\perp\right| \leq \|W\pi^\perp\| \leq \omega.$$

Thus,

$$\left|\pi^\top W\pi\right| \leq \varepsilon_\tau^2 + \omega \leq \varepsilon_0^2 + \omega.$$

□

Going forward, we fix a nonzero linear test $\tau \in \{0,1\}^n$ once and for all and, for simplicity, write ε_0 (the starting small-bias guarantee) instead of ε_τ. Having fixed τ, it is convenient to introduce the diagonal matrix Π whose diagonal encodes π. Thus, the bias[3] of the resulting set (with respect to τ), $\pi^\top W\pi$, can be written as

$$\mathbf{1}^\top\Pi W\Pi\,\mathbf{1}. \tag{2}$$

In what follows—especially as the constructions become more involved—it will be helpful to reason with diagrams that track the "flow" of a vector, decomposed into several subspaces, as operators act on it. For example, in the proof of Lemma 2.1, in light of Equation (2), we wish to understand how the component in span$\{\mathbf{1}\}$—and in particular its norm—evolves as we apply the three operators Π, W, Π, and then project back onto that

[3] In this section we do not distinguish between bias and signed bias.

space.

This behavior is illustrated in Figure 1. The top row, labeled $\|$, represents vectors in $\mathrm{span}\{\mathbf{1}\}$; the bottom row, labeled $\perp$, represents vectors in $\mathbf{1}^{\perp}$. The columns correspond to the operators being applied and should be read right-to-left, matching the order of operator multiplication when acting on column vectors. From each node there are (typically) two outgoing edges, indicating to which subspace the operator maps the vector. For example, if only one horizontal edge appears, the corresponding subspace is invariant under that operator. A number on an edge denotes the contraction factor guaranteed on the projection to the target subspace. An unlabeled edge indicates no quantitative bound; however, all operators we consider have operator norm at most 1, so the norm does not increase.

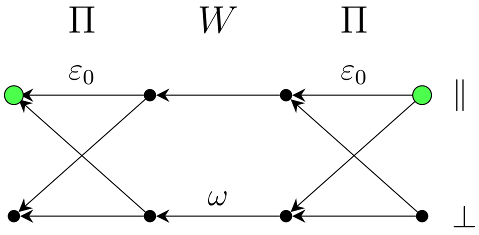


Figure 1: Illustration of the Rozenman–Wigderson bias reduction via expander-edge sampling.

Using this diagram, one can *see* the $\varepsilon_0^2 + \omega$ bound proved in Lemma 2.1 by summing—over all paths from the right green circle to the left green circle—the product of the edge weights along each path.

Let us instantiate the Rozenman–Wigderson bias reduction procedure with a concrete expander. In particular, take a Ramanujan graph, which satisfies $\omega \approx \frac{2}{\sqrt{d}}$, and—by Lemma 2.1—choose the spectral parameter so that $\omega = \varepsilon_0^2$. A quick calculation then shows that, with this choice, the procedure transforms[4] an ε-biased set of size n/ε^c, for some constant c, into an ε-biased set of size

$$O\left(n \cdot \left(\frac{1}{\varepsilon}\right)^{4+c/2}\right).$$

Thus, iterating the procedure yields ε-biased sets with size $n/\varepsilon^{4+o_\varepsilon(1)}$.

2.1.1 Taking longer walks

A perhaps even more natural approach is to take longer walks on the expander graph discussed above. In particular, one might take two steps on G (which corresponds to

[4] More precisely, as is standard in asymptotic analysis, we apply the procedure to a family of small-bias sets that (ideally) contains a set for every choice of n and ε.

taking the parity of three vertices rather than two). Extending Equation (2) to this longer walk, one can show that the bias of the resulting set is

$$\mathbf{1}^\top \Pi W \Pi W \Pi \, \mathbf{1}. \tag{3}$$

Unfortunately, as depicted in Figure 2, the second G step is effectively "wasted." This is because we have very little control over the operator Π (which is induced by the linear test). In particular, Π may map a vector in $\mathbf{1}^\perp$ largely into $\mathbf{1}^\parallel$ (see the bold red edge in the diagram), rendering the subsequent G step moot.

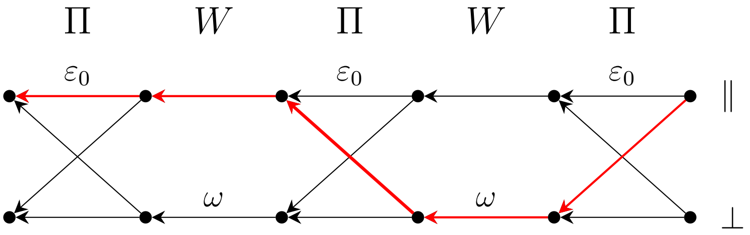


Figure 2: Illustration of how the second step is "wasted" by the interleaved operator Π.

A quick calculation shows that, when we take a long walk, losing every other G-step—together with the unavoidable quadratic dependence of the spectral expansion parameter ω on the degree d of the expander—again yields a sample space of size about n/ε^4.

2.1.2 Ta-Shma's approach

The difficulty with taking long walks, as discussed in Section 2.1.1, is that the "adversarial" operator Π can interfere with the random walk on G in an uncontrolled way—mixing the subspaces $\mathrm{span}(\mathbf{1})$ and $\mathbf{1}^\perp$ and thereby disrupting the walk on G. As hinted in the introduction, the key insight underlying Ta-Shma's improvement is to restrict, at each step, the set of neighbors a vertex of G may transition to in a manner that is "hidden" from the adversary. In this way, the effect of Π on the walk remains largely under our control.

Ta-Shma implements this approach using the s-wide replacement product, introduced by Ben-Aroya and Ta-Shma [BATS11] for the purpose of constructing near-Ramanujan graphs. Before proceeding, we recall the definitions of the replacement product and the wide-replacement product.

2.2 The s-wide replacement product

2.2.1 The replacement product

Let G be an undirected d-regular graph with spectral expansion ω_G. For each vertex $v \in V(G)$ we arbitrarily label its d neighbors by indices in $[d] = \{1, \ldots, d\}$. Thus, for $i \in [d]$ we may speak of the i-th neighbor of v. If this neighbor is u, then, in general, u may regard v as its j-th neighbor for some $j \in [d]$; the labelings need not be consistent, so j need not equal i. We refer to this labeling as the *rotation map* of G.

Let H be an undirected graph on d vertices with vertex set $V(H)$, which we identify with $[d]$ and use interchangeably. Define the *replacement product* graph $G \circledR H$ on the vertex set $V(G) \times V(H)$ as follows. For each $v \in V(G)$, place a copy of H on the vertices $\{(v, i) : i \in V(H)\}$; we call this set the *cloud* of v. In addition, connect (v, i) to (u, j) whenever u is the i-th neighbor of v and v is the j-th neighbor of u. Edges within a cloud are called *intra-cloud* edges, and edges connecting distinct clouds are called *inter-cloud* edges.

The replacement product serves as a basic building block for more elaborate constructions, such as the zig–zag product [RVW00], a key tool in constructing expander graphs. To obtain expanders with improved parameters, Ben–Aroya and Ta–Shma introduced a variant of the replacement product, which we now describe.

2.2.2 The s-wide replacement product

Let $s \geq 1$ be a parameter, called the *width*. For the purposes of bias reduction, it is helpful to view the s-wide replacement product not primarily as a graph construction but as a particular random walk on a graph.

Let G be a graph as in Section 2.2.1. The first departure from the standard replacement product is that we now take a graph H on d^s vertices. As in prior applications for constructing expanders we would like both G and H to be good expanders. However, for the purposes of the s-wide replacement product, we also require additional structure on H.[5] More precisely, let Λ be an arbitrary abelian group of order d, and identify $[d]$ with Λ. We then take H to be a Cayley graph on the product group Λ^s with respect to some fixed symmetric generating set C of size c.

The vertex set of the graph on which we are going to take a walk is $V(G) \times V(H)$. As in the discussion of the replacement product, we view the set $\{(v, h) : h \in V(H)\}$ as the *cloud of* v. We now describe length-s walks.

Fix a starting vertex $(v_0, h_0) \in V(G) \times V(H)$ and a sequence of indices $i_1, \ldots, i_s \in [c]$ (each i_t selects a neighbor in H). For $t = 1, \ldots, s$ perform:

[5] Interestingly, the original version of [BATS11] analyzed a random H; in a later version, imposing structure on H enabled a cleaner analysis.

1. Let h_t be the i_t-th neighbor of h_{t-1} in H.
2. Write $h_t = (h_t^{(1)}, \ldots, h_t^{(s)}) \in \Lambda^s$, and let v_t be the $h_t^{(t)}$-neighbor of v_{t-1} in G (recall that we identify Λ with $[d]$).

After phase t the state is (v_t, h_t). In particular, phase t uses the t-th coordinate, or "register", of h_t to determine the step on G.

To implement Ta–Shma's idea of "hiding" the random walk from the adversarial operator Π, we label *clouds*—rather than individual vertices—by ± 1, using the signs induced by an ε_0-biased set with respect to a fixed linear test. In other words, we identify the initial small-bias set with the set of clouds $V(G)$ (rather than with the full vertex set $V(G) \times V(H)$). Equivalently, for every $v \in V(G)$ we assign the same label to all vertices (v, h) with $h \in V(H)$.

The combinatorial idea above takes a particularly clean form in the operator viewpoint. The three operators we now introduce act on vectors indexed by $V(G) \times V(H)$ (equivalently, on $\mathbb{R}^{V(G)\times V(H)}$).

- **Adversarial operator.** Because it is indifferent to the structure *within* each cloud, it factors as
$$\Pi \otimes I_{|V(H)|},$$
where we write $\otimes$ for the Kronecker product, and $I_{|V(H)|}$ is the identity on the $|V(H)|$ coordinates inside a cloud. For notational convenience, we overwrite the symbol and denote this lifted operator again by Π.

- **Intra-cloud operator.** The operator H acts identically within each cloud, independent of which cloud we are in. Hence it factors as
$$\widetilde{H} \triangleq I_{|V(G)|} \otimes H.$$
We emphasize that Π and $\widetilde{H}$ act on different registers. This is the operator-level manifestation of the combinatorial "hiding" idea: H dictates the within-cloud motion, while the adversary Π can only affect the other register.

- **Inter-cloud operators.** Finally, we have the operators corresponding to G. They encode the rotation map of G and keep track of the step index $i \in [s]$, which determines which register is used. Using the product structure $V(H) = \Lambda^s$, we view the H part as s registers. For each $i \in [s]$, define $\dot{G}_i$ to encode the rotation map while consulting only the i-th register of the H-register. Concretely, $\dot{G}_i$ is a permutation that (i) moves in the $V(G)$ coordinate according to the value stored in the i-th register, and (ii) updates that i-th register to the return label specified by the rotation map; all other H-registers remain unchanged.

2.3 Analysis

We now sketch the analysis of Ta-Shma's construction via the s-wide replacement product, starting with the special case $s = 2$. Our state space consists of labelings of the graph with vertex set $V(G) \times V(H)$, so we consider the function space

$$X \triangleq \mathbb{R}^{V(G)\times V(H)}.$$

Since there are $s = 2$ H-registers, together with the G-register we have three registers in total. Decomposing each register into $\mathrm{Span}(\mathbf{1})$ and its orthogonal complement $\mathbf{1}^{\perp}$, we obtain a direct-sum decomposition of X into eight subspaces:

$$X \;=\; V_{\emptyset}^{\|} \,\oplus\, V_{1}^{\|} \,\oplus\, V_{2}^{\|} \,\oplus\, V_{12}^{\|} \,\oplus\, V_{\emptyset}^{\perp} \,\oplus\, V_{1}^{\perp} \,\oplus\, V_{2}^{\perp} \,\oplus\, V_{12}^{\perp}. \tag{4}$$

By our notation, the superscript indicates whether the G-register lies in $\mathrm{Span}(\mathbf{1})$ ("$\|$") or in $\mathbf{1}^{\perp}$ ("$\perp$"). The subscript records which of the two H-registers (if any) lie in $\mathbf{1}^{\perp}$. Here $\mathbf{1}$ denotes the all-ones vector in the relevant register.

In parallel with Equation (3), the bias of the linear test encoded by Π is now

$$\mathbf{1}^{\top}\Pi\,\underbrace{\dot{G}_2\widetilde{H}\Pi}_{M_2}\,\underbrace{\dot{G}_1\widetilde{H}\Pi}_{M_1}\,\mathbf{1}. \tag{5}$$

We trace the effect of the operators from right to left (see Figure 3). First, write $\mathbf{1}$ as $\mathbf{1}\otimes\mathbf{1}\otimes\mathbf{1}$, ordered by the three registers—G first, followed by the two H registers—so $\mathbf{1} \in V_{\emptyset}^{\|}$. Now:

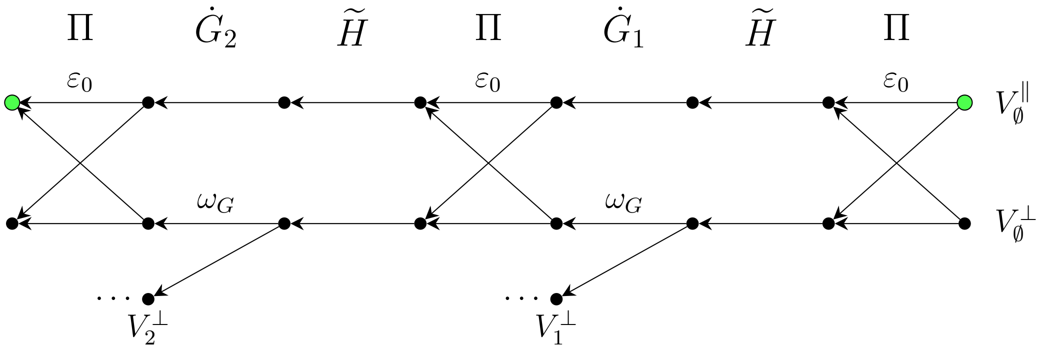


Figure 3: Bias evolution in the $s = 2$ wide–replacement product.

1. The first operator Π acts only on the G register and decomposes the vector into two components: an ε_0 fraction that remains in $V_{\emptyset}^{\|}$, and a dominant component that moves to $V_{\emptyset}^{\perp}$ (i.e., the G coordinate lies in $\mathbf{1}^{\perp}$).
2. The next operator $\widetilde{H}$ is agnostic to the G register and acts only within the H registers.

Since the H part is $\mathbf{1}$ (as a vector of length $|V(H)|$), $\widetilde{H}$ leaves it unchanged.

3. The third operator applied is $\dot{G}_1$, which leaves the $V_\emptyset^{\|}$ component unchanged. For the $V_\emptyset^{\perp}$ component, only an ω_G fraction of the mass remains in this space; the rest is moved to $V_1^{\perp}$.

   The key point is that register 1 of H will not be used again. Therefore, from this point on in this branch, every $\widetilde{H}$ step acts on a vector in $\mathbf{1}^{\perp}$ (as a vector of length $|V(H)|$), so that component shrinks by a factor of ω_H at each step (in short example, in the remaining second step). We therefore stop tracking it in the diagram, and consider it as a "win".

   We emphasize at this point that our construction departs significantly from Ta-Shma's: we reuse the same register more than once—indeed, *exponentially* many times (in the number of registers). This reuse of "compromised" registers follows a specific pattern and requires a more delicate analysis.

4. Now comes the second action of Π. We've already analyzed what it is doing on the component in $V_\emptyset^{\|}$ (but for an ε_0 mass, it moves to the $V_\emptyset^{\perp}$). However, on the $V_\emptyset^{\perp}$ component we have no control over Π's adversarial behavior in the sense that it can move all mass back to $V_\emptyset^{\|}$. What we can say - and this is the key property - is that Π does not touch the H component at all (and so we are still within the $\emptyset$ subscript).

5. As in the previous application of $\widetilde{H}$, here too $\widetilde{H}$ ignores the G register and acts only on the H registers. Since we consider only branches in which the H part is $\mathbf{1}$, $\widetilde{H}$ leaves it unchanged.

6. As before, $\dot{G}_2$ leaves the $V_\emptyset^{\|}$ component unchanged and shrinks the $V_\emptyset^{\perp}$ component by a factor of ω_G, moving most of it to $V_2^{\|}$.

7. Finally, the operator Π behaves as before: it shrinks the $V_\emptyset^{\|}$ component by ε_0 and may move $V_\emptyset^{\perp}$ back to $V_\emptyset^{\|}$.

Thus, we can bound the bias in Equation (5) by summing over all branches that start and end in $V_\emptyset^{\|}$ (the green nodes in Figure 3), yielding

$$\varepsilon_0^3 + 2\varepsilon_0\,\omega_G + \omega_G^2.$$

Setting $\omega_G = \varepsilon_0$ simplifies this to $O(\varepsilon_0^2)$.

The bound obtained above may not seem like much: indeed, a 2-wide replacement product gives no advantage over plain random walks, as discussed in Section 2.1.1 for bias reduction. The advantage of the s-wide replacement product kicks in only as we increase s. The key feature is that once we move beyond the two subspaces $V_\emptyset^{\|}$ and $V_\emptyset^{\perp}$, we obtain a shrink by ω_H at every $\widetilde{H}$ step. However, to analyze Ta-Shma's construction we also have to consider bilinear forms beyond the specific quadratic form of Equation (5), corresponding to start/end states other than $V_\emptyset^{\|}$. In those cases additional contributions involving ω_H arise: Loosly speaking, once the adversary leaves the $\emptyset$ spaces (that is, $V_\emptyset^{\|}$, $V_\emptyset^{\perp}$) at every step the vector contracts by a factor of ω_H. When returning to the the $\emptyset$ spaces, in every other step a contraction of ω_G or ε_0 occurs.

Formalizing this argument, one can show that there exist universal constants a, b—in particular, independent of s—such that, setting $\omega_G = \varepsilon_0 = \omega_H^a$, we have

$$\left\|M_s M_{s-1} \cdots M_1\right\| \;=\; O(\omega_H)^{s-b}\,.$$

For ease of exposition we will assume here $a = b = 1$, as the exact constants are unimportant. In particular, we will simply use the bound ω_H^{s-1}.

With this in hand, consider a bias-reduction procedure that takes a length-st random walk, cycling through the s registers periodically, for a parameter t that we will optimize shortly. For simplicity, assume both H and G are Ramanujan graphs. This is, strictly speaking, an idealized assumption—H must satisfy structural constraints that preclude Ramanujan optimality—but it streamlines the exposition and affects the parameters only mildly at this point (though it will become the main remaining challenge for achieving optimal small-bias sets beyond our work). Under this assumption, the bias after the procedure is bounded by

$$\varepsilon \;=\; \omega_H^{(s-1)t}.$$

The size of the resulting ε-biased set is

$$|V(G)| \cdot |V(H)| \cdot c^{st} = O\left(\frac{n}{\varepsilon_0^4} \cdot d^s \cdot c^{st}\right). \tag{6}$$

Substituting ω_H for ε_0 and ω_G, and using the relations between the expander degrees and spectral expansions, we obtain (after a slight oversimplification that does not affect the asymptotics) a set of size

$$\frac{n}{\varepsilon^2} \cdot 2^{st} \cdot \left(\frac{1}{\varepsilon}\right)^{\frac{1}{s}+\frac{1}{t}}.$$

The factor multiplying the target size $\frac{n}{\varepsilon^2}$ is symmetric in s and t, so to minimize it we take $s = t$. The minimum is then attained at $s = (\log \frac{1}{\varepsilon})^{1/3}$, resulting in an ε-biased set of size

$$O\left(\frac{n}{\varepsilon^2} \cdot 2^{\left(\log \frac{1}{\varepsilon}\right)^{2/3}}\right).$$

This matches Equation (1) up to a $2^{\mathrm{poly}(\log\log\frac{1}{\varepsilon})}$ factor, stemming from the fact that we cannot take H to be Ramanujan.

3 Proof Overview

As described in Section 2, Ta-Shma's approach to bias reduction follows the Rozenman–Wigderson random-walk framework, but chooses its steps using s registers that are hidden from the adversary (the linear test we seek to fool), encoded by the operator Π. When optimizing the parameters, we found in Section 2.3 that it is best to consider a walk of length s^2, analyze it in blocks of length s, and then apply the submultiplicativity of the spectral norm to bound the entire product. Specifically, we partition the walk into s phases, each of length s; in each phase we "restart" the analysis and treat the s registers as fresh (unused in that phase).

Our main technical contribution is to show that these s registers can be reused—even after they have been "compromised"—provided this is done carefully. To illustrate our approach, we first analyze the simplest setting with no adversary; that is, the operator Π does not appear in the product. While the s-wide replacement product as used for bias reduction is not meaningful under this assumption, it is interesting in its own right to reuse registers in the s-wide replacement product for expander constructions—indeed, this was Ben-Aroya and Ta-Shma's original motivation for introducing the wide product [BATS11]. However, we begin by ignoring Π mainly for expository clarity: it is easier to present some of the core ideas in this simplified setting and then introduce the additional ideas needed to handle the presence of Π. In the next section we first consider this setting in the special case $s = 2$.

3.1 The non-adversarial setting

Since Π is absent, we consider products of the form $M_3M_2M_1$ with $M_i = \dot{G}_i\widetilde{H}$. We wish to understand whether registers can be reused; for example, we may study expressions such as $M_1M_2M_1$, in which the first register is reused. To this end, we take a closer look at how the operators $\widetilde{H}$, $\dot{G}_1$, and $\dot{G}_2$ act on the subspaces $V_S^{\perp}$ and $V_S^{\parallel}$ for all $S \subseteq \{1, 2\}$. Figure 4 summarizes the behavior of $\widetilde{H}$ (we prove this formally in Section 4.2). In particular, whenever $S \neq \emptyset$, we pick up a factor of ω_H. Hence, from the adversary's perspective, the goal is to "revisit" $V_\emptyset^{\perp}$ or $V_\emptyset^{\parallel}$ as often as possible so as to avoid this contraction to the bias. Moreover, note that $\widetilde{H}$ does not move us between components.

As we prove in Section 4.4, the operator $\dot{G}_1$ acts as follows (see Figure 5). On the parallel subspaces, $\dot{G}_1$ leaves us in place. On the perpendicular subspaces, $\dot{G}_1$ moves only along the axis corresponding to index 1: it maps between $V_\emptyset^{\perp} \leftrightarrow V_{\{1\}}^{\perp}$ and $V_{\{2\}}^{\perp} \leftrightarrow V_{\{1,2\}}^{\perp}$. Moreover, if we start in $V_\emptyset^{\perp}$ or $V_{\{2\}}^{\perp}$, at most an ω_G fraction of the norm remains in the original subspace. The operator $\dot{G}_2$ acts in the same manner with 1 replaced by 2.

Think of S as the set of *compromised* registers, i.e., registers with which we cannot perform a random-walk step on G. A register in state $\mathbf{1}$ simulates a truly random step on

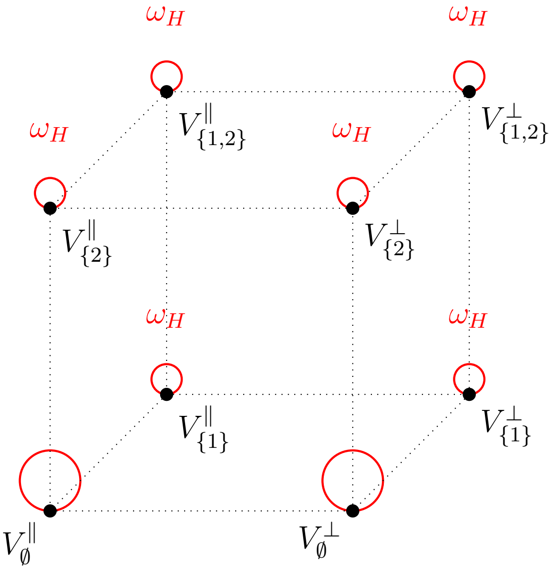


Figure 4: Illustration of the action of $\widetilde{H}$.

G, while a register in $\mathbf{1}^{\perp}$ yields an invalid step. However, if we are in the G-parallel register $V^{\|}_S$, the choice of step is immaterial.

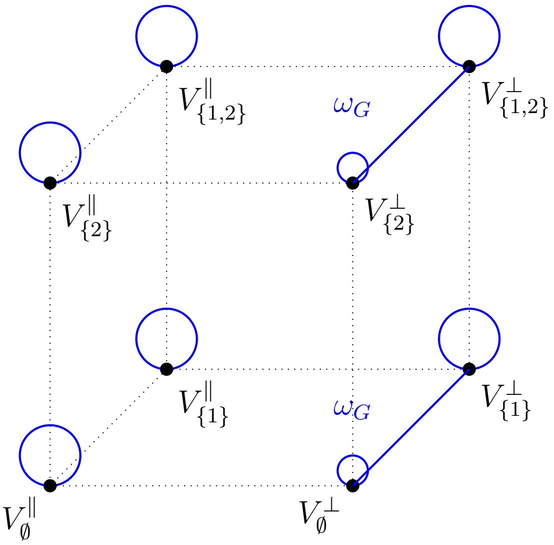


Figure 5: An illustration of the action of $\dot{G}_1$.

3.1.1 The pebble game

A useful way to think about reusing registers is as a pebble game on the s-dimensional hypercube whose vertices are indexed by subsets $S \subseteq [s] \triangleq \{1, \ldots, s\}$. We will start with the simplest case $s = 2$; the extension to general s is discussed afterwards.

At the start of the game, the opponent[6] places a single pebble at $\emptyset$. At each step the adversary moves the pebble according to rules specified below. In some cases the pebble may duplicate; when multiple pebbles are present, all pebbles move simultaneously according to the same rules. Whenever several pebbles are places at the same node, they merge to one. The game lasts t steps, and the adversary's objective is to have a pebble at $\emptyset$ for as many steps as possible.

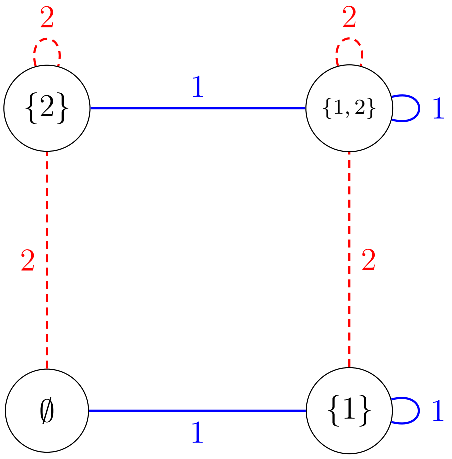
The opponent plays against us—the bias-reduction designer—who chooses a sequence in $\sigma \in [s]^t$ (this corresponds to the order in which registers are used) so as to minimize the opponent's gain. The rules of the game are as follows: Let $\sigma_k \in [s]$ and consider a pebble placed at node S:

- If $\sigma_k \in S$, the pebble stays put but *duplicates*: the copy is placed at $S \setminus \{\sigma_k\}$.
- If $\sigma_k \notin S$, the pebble moves to $S \cup \{\sigma_k\}$.

These rules describe the dynamics in the G-perpendicular registers. The parallel registers will come into play only after we reintroduce the adversary, so for now we restrict attention to the G-perpendicular component. We initialize with mass concentrated at $V_{\emptyset}^{\perp}$, thereby giving the opponent a free win in the first step.

The first rule reflects that when we use a compromised register at time k (i.e., $\sigma_k \in S$), we have no control over the corresponding G operator, and the mass may either remain in $V_S^{\perp}$ or move to $V_{S\setminus\{\sigma_k\}}^{\perp}$. The second rule corresponds to using a "good" register; in this case, most of the mass moves to $V_{S\cup\{\sigma_k\}}^{\perp}$. Here we ignore the ω_G-contracted mass that stays in place. In the full analysis we account for it as well.

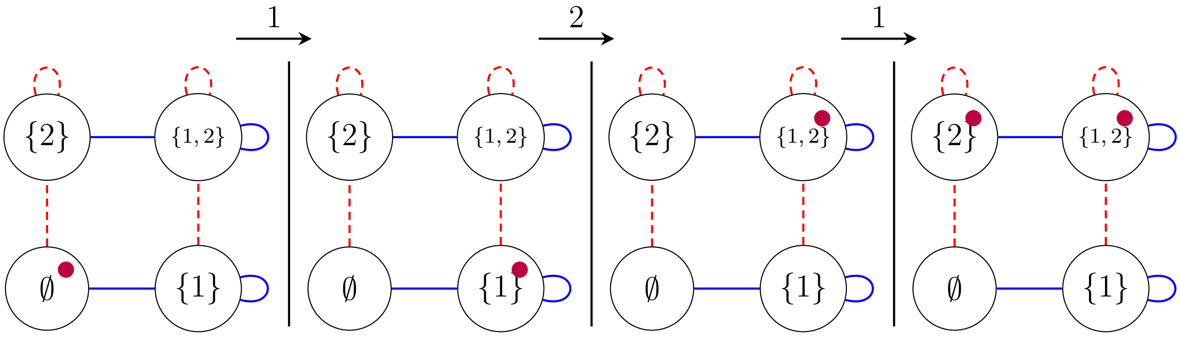
The following figure captures the rules of the game in a diagram.



Let us play! In this game we are playing a length 3 game corresponding to the string 1 2 1 capturing the reuse of register 1. As the following diagram depicts, in this game we

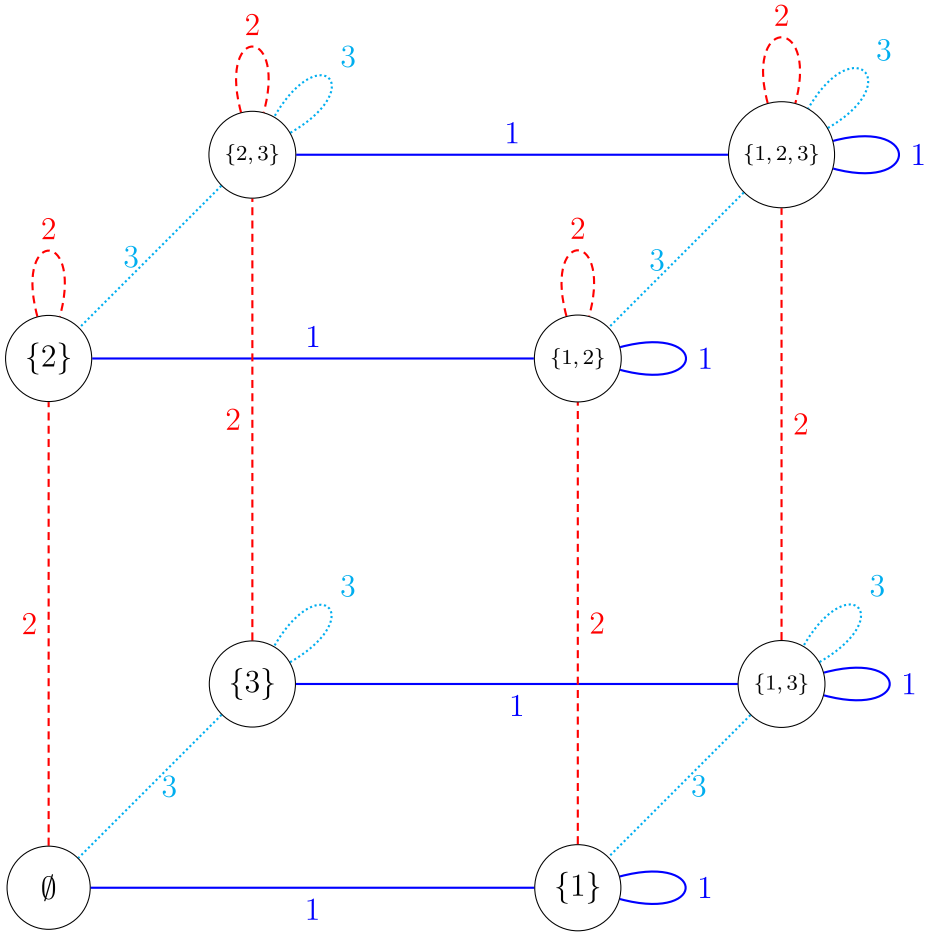
[6] In this game we call the player we face the *opponent*; we reserve the term *adversary* for the entity that fixes Π.

are able to make this re-usage work. Indeed, the opponent was not able to place a pebble at $\emptyset$ except for the initial position.



The careful reader may notice that the string $1\,2\,2\,2\,2\cdots$ also works. This is a consequence of an expository simplification: we analyzed only a particular quadratic form, whereas the full argument requires handling additional bilinear forms. In the language of the pebble game, the correct objective is to win on every interval, not just on the entire word.

The following diagram illustrates the extension to width $s = 3$. From the figure we see that playing the register sequence $1\,2\,1\,3\,1\,2\,1$ again prevents the opponent from placing a pebble at $\emptyset$ after the first step. In this three-register case, register 1 is used every other step.



This generalizes to s registers via the flip-index sequence of the binary-reflected Gray code on s coordinates, defined recursively by

$$F_1 = (1), \qquad F_s = (F_{s-1},\, s,\, F_{s-1}),$$

where juxtaposition denotes concatenation. Note that the sequence has length $2^s - 1$.

Using this sequence yields an ω_H-factor contraction of the bias at every step except the first. Thus far, we have ignored the possibility that the opponent "cheats," i.e., deviates from the rules by paying a cost ω_G. Nevertheless, we show that the sequence is robust to such strategies. In particular, when ω_G is only modestly smaller than ω_H—say $\omega_G = \omega_H^c$ for some universal constant c independent of s—the opponent has no incentive to cheat. To be somewhat more precise, the opponent may be in a superposition of "cheating" and "not cheating"; the claim above should be understood as conveying the intended intuition.

3.2 The return of the adversary

We now reintroduce the adversary. The game is now played on two s-dimensional hypercubes: the G-perpendicular cube and the G-parallel cube. The simplified game above was restricted to the perpendicular cube—whose node labels we now denote by $S^{\perp}$—and its rules remain unchanged. The rules on the parallel cube are simpler and can be deduced from Figures 4 and 5: every move is a self-loop, i.e., the pebble stays put. The opponent's objective is to maximize the number of time steps with a pebble at $\emptyset^{\perp}$ or $\emptyset^{\|}$. As before the game begins with a single pebble at $\emptyset^{\perp}$.

The key new feature is that after each instruction σ_k (the register chosen at time k), a Π-step occurs that may move pebbles between the two cubes, thereby giving the opponent additional freedom. To determine the rules for the Π-steps, we first analyze the Π operator. Figure 6 illustrates its behavior. Starting from $V_S^{\|}$, Π keeps only an ε-fraction of the norm in $V_S^{\|}$. The self-loops and the horizontal arrow are the only outgoing edges from $V_S^{\|}$ and $V_S^{\perp}$. For simplicity of exposition, we ignore the ε-contracting steps (much as we ignored the ω_G-contracting steps earlier). With this, the Π-steps follow these rules:

- A pebble at $S^{\|}$ moves to $S^{\perp}$.
- A pebble at $S^{\perp}$ remains in place and duplicates, creating a new pebble at $S^{\|}$.

We summarize all rules in Figure 7.

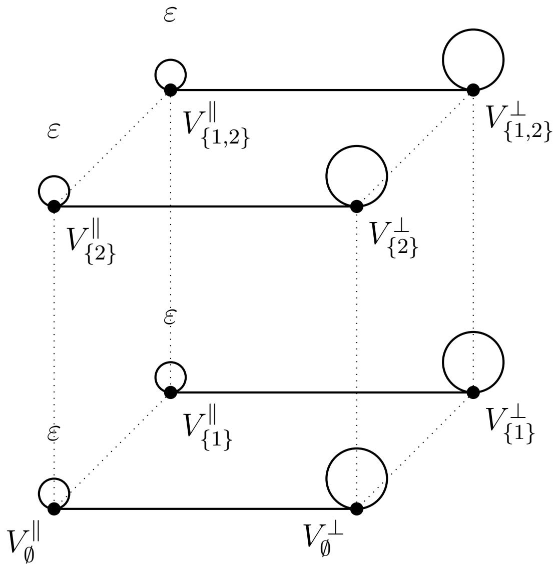


Figure 6: The action of Π on the spaces $V_S^{\perp}$, $V_S^{\|}$.

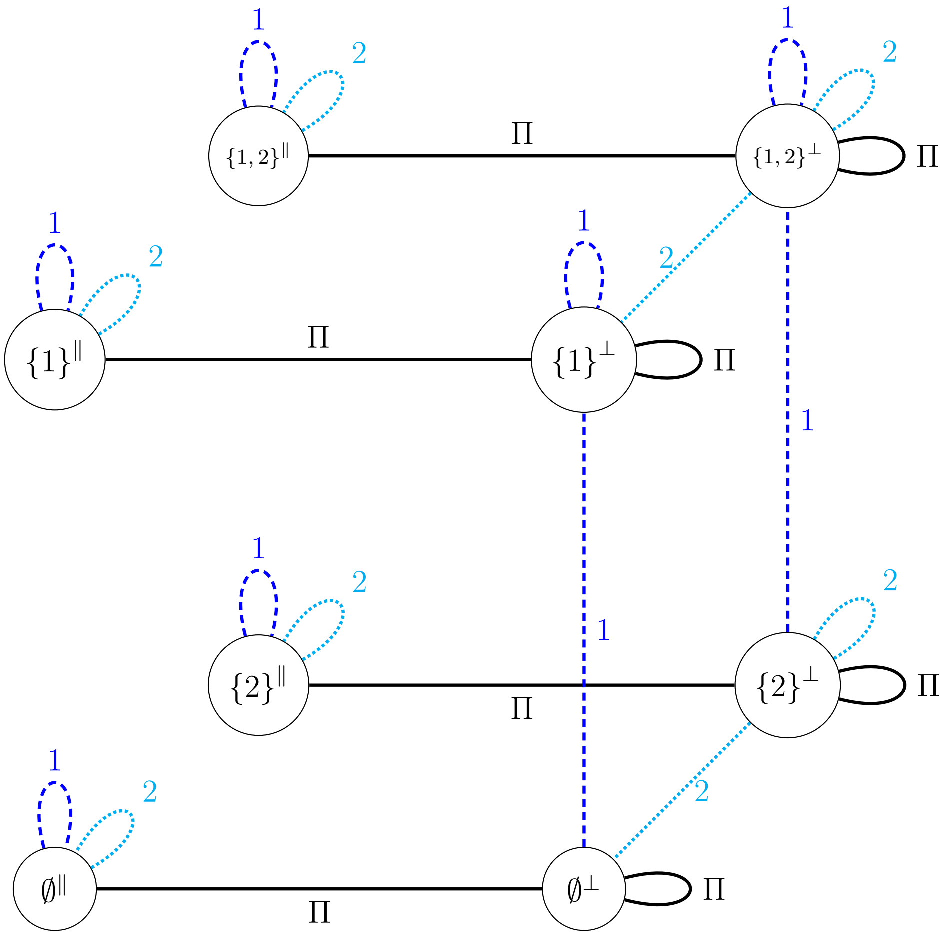


Figure 7: A complete illustration of the two-register pebble game with Π.

Let us now play the same $1\,2\,1$ string as before, but interleaved with Π-steps: $\Pi\,1\,\Pi\,2\,\Pi\,1$. The reader may verify that, because of the Π-steps, the number of time steps with a pebble in the $\emptyset$-states is now two, compared with only one in the setting without Π-steps. While the difference between one and two may not seem compelling, the failure of the Gray-code strategy becomes much more evident as s increases.

It turns out that with two registers there is no way to reuse registers. The Gray-code string $1\,2\,1\,3\,1\,2\,1$ (with interleaved Π-steps, omitted for readability) from before also fails for three registers. However, the following string is a good play: $1\,2\,3\,1$—using it, the opponent cannot return to the $\emptyset$ states. With five registers, one can achieve the same effect with the string

$$1\,2\,3\,1\,4\,5\,1\,2\,3\,1.$$

In general, we find a string on s registers of length $\approx 2^{s/2}$ that guarantees the opponent never revisits the $\emptyset$ states. The factor-2 loss in the exponent relative to the no-Π game is immaterial.

We emphasize that matters become more involved once we reintroduce the "cheats" the opponent can perform via ω_G-steps and ε_0-steps (i.e., the Π-steps). Nevertheless, by choosing ε_0 in the same manner as ω_G, we prove the opponent has no incentive to cheat. We summarize the entire two-register game in Figure 8.

3.3 Setting parameters

We now instantiate our new bias-reduction procedure with s registers (hence walks of length $\ell = 2^{s/2}$), starting from an ε_0-biased set of size $O(n/\varepsilon_0^4)$ constructed in the previous sections. The size of the resulting ε-biased set is

$$|V(G)| \cdot |V(H)| \cdot c^{2^{s/2}} = O\left(\frac{n}{\varepsilon_0^4} \cdot d^s \cdot c^{2^{s/2}}\right). \tag{7}$$

Compare this with Equation (6), where the walk length is exponentially shorter than in our setting.

Substituting ω_H for ε_0 and ω_G for simplicity, as chosen in the previous sections, and using the relations between expander degrees and spectral parameters, we obtain (after a slight oversimplification that does not affect the asymptotics)

$$\frac{n}{\varepsilon^2} \cdot 2^\ell \cdot \left(\frac{1}{\varepsilon}\right)^{\frac{\log \ell}{\ell}}. \tag{8}$$

The loss term is minimized at $\ell \approx \sqrt{\log \frac{1}{\varepsilon}}$, yielding an improvement over Ta-Shma's construction—namely, an ε-biased set of size

$$\frac{n}{\varepsilon^{2+\alpha}}, \qquad \alpha = \widetilde{O}\left(\left(\log \tfrac{1}{\varepsilon}\right)^{-1/2}\right),$$

as stated in Theorem 1.1.

3.4 Toward optimal small-bias sets

The reason we do not obtain optimal small-bias sets is the need to balance the two terms in Equation (8)—namely, 2^ℓ, which increases with ℓ, and $(1/\varepsilon)^{\frac{\log \ell}{\ell}}$, which decreases. The first "exponential in ℓ" term does not in fact have base 2 but rather some constant (and in fact, slightly super-constant); we chose to present it as if it is 2 in this informal overview for simplicity. By inspection, this constant comes from two sources:

- The inherent factor of 2 in the relation between spectral expansion and degree, $\omega \approx \frac{2}{\sqrt{d}}$. This was also one of the bottlenecks noted by Ta-Shma in improving his construction.

- A deliberately lossy step in our bias bound: for simplicity we sacrificed absolute constants, which are not currently the bottleneck.

Moving forward, we will need to eliminate both sources of loss. We believe the first is more inherent, while the second is largely technical. As noted, the factor of 2 is not even precise, since the expander H needed for our bias-reduction procedure has additional structure that prevents it from being Ramanujan. Even setting this aside, we still wish to avoid this factor of 2. We believe this may be achievable by switching to a non-backtracking version of our procedure or—essentially equivalently—by using a directed graph H. The main difficulty is to combine such a modification with the machinery developed in this work.

An alternative is to use rotating expanders [CM23], which keep us in the realm of undirected graphs. Although each individual expander carries a factor-2 penalty, suitable compositions avoid an exponential blow-up and incur only a linear loss. Here too, the challenge is to integrate our framework with rotating expanders.

4 A Formal Framework for the Wide-Replacement Product

In this section, we formalize the mechanism underlying the wide-replacement product. Specifically, we present a variant of the original construction of Ben-Aroya and Ta-Shma [BATS11]. Our presentation is tailored to our needs: it is more abstract, yet it imposes additional constraints. Moreover, we incorporate into the product an auxiliary operator Π. This operator is introduced precisely to enable an analysis of the wide-replacement product in the setting of small-bias sets, following [TS17].

Ingredients

The components of the wide-replacement product are as follows and will remain fixed throughout this section:

- Let Λ be a finite group of size D.
- For a parameter s, dubbed the *width* parameter, let H be the Cayley graph over the group Λ^s corresponding to a generating set $C \subseteq \Lambda^s$; that is, $H = \mathrm{Cay}(\Lambda^s, C)$. We denote the spectral expansion of H by ω_H.
- Let G be a D-regular graph on n vertices, and write ω_G for its spectral expansion. We represent G via a rotation map

$$\mathrm{Rot} : V(G) \times \Lambda \to V(G) \times \Lambda.$$

We assume that the rotation map is locally invertible, i.e., there exists a function $\phi : [D] \to [D]$ such that for every $g \in V(G)$ and $x \in [D]$,

$$\mathrm{Rot}(g, x) = (h, \phi(x))$$

for some $h = h(g, x) \in V(G)$. In words, the second component of the output depends only on the second component of the input. We slightly abuse notation by sometimes considering only the projection of the rotation map onto the first register.

The space X

We denote the $\mathbb{R}$-vector space of all real functions on Λ by V, that is,

$$V \triangleq \{f : \Lambda \to \mathbb{R}\}.$$

We similarly define

$$W \triangleq \{f : V(G) \to \mathbb{R}\},$$

where $V(G)$ is the vertex set of G. Define

$$X \triangleq V^{\otimes s} \otimes W. \tag{9}$$

We write $V^{\otimes s}$ as $V_1 \otimes \cdots \otimes V_s$, where each $V_i \cong V$ as an $\mathbb{R}$-vector space. We refer to V_i as the *i-th register* and to the entire factor $V^{\otimes s}$ as the *H-register*. We refer to the factor W in the tensor product $X = V^{\otimes s} \otimes W$ as the *G-register*.

Decomposing the space X

For an $\mathbb{R}$-vector space U, there is a natural decomposition into parallel and perpendicular subspaces:

$$U^{\parallel} \triangleq \mathrm{span}\left\{\mathbf{1}\right\}, \qquad U^{\perp} \triangleq \left(U^{\parallel}\right)^{\perp},$$

where $\mathbf{1}$ is the all-ones vector, thus $U = U^{\parallel} \oplus U^{\perp}$. Note that we slightly abuse notation: here $U^{\perp}$ denotes the orthogonal complement of $U^{\parallel}$ *inside* U (with respect to the standard inner product), rather than the orthogonal complement of U in some ambient space; this convention is convenient and will not cause ambiguity.

Recall that the tensor product distributes over direct sums, namely

$$(U_1 \oplus U_2) \otimes U_3 \cong (U_1 \otimes U_3) \oplus (U_2 \otimes U_3).$$

Applying this repeatedly yields a decomposition of the space X from Equation (9) into 2^{s+1} subspaces: in each coordinate we choose either the parallel or the perpendicular component (for each of $V_1, \ldots, V_s$ and for the G-register). For a set $S \subseteq [s]$, we denote

$$V_S \triangleq \bigotimes_{i=1}^{s} U_i, \qquad \text{where} \qquad U_i \triangleq \begin{cases} V_i^{\perp}, & i \in S, \\ V_i^{\parallel}, & i \notin S. \end{cases}$$

Equivalently (as a shorthand), we may write

$$V_S = \bigotimes_{i \in S} V_i^{\perp} \, \otimes \, \bigotimes_{i \notin S} V_i^{\parallel},$$

with the tacit convention that tensor factors are ordered by increasing index. Thus, for example, if $S = \{1, 3\} \subseteq [3]$, then the display above yields $V_S = V_1^{\perp} \otimes V_3^{\perp} \otimes V_2^{\parallel}$, which we identify with $V_1^{\perp} \otimes V_2^{\parallel} \otimes V_3^{\perp}$ by reordering the factors.

Note that V_S itself is not a subspace of X, as it does not include the W factor. To obtain subspaces of X, define

$$V_S^{\parallel} \triangleq V_S \otimes W^{\parallel}, \qquad V_S^{\perp} \triangleq V_S \otimes W^{\perp}, \qquad V_S^{\perp\!\!\perp} \triangleq V_S \otimes W. \tag{10}$$

Thus, the superscript in $V_S^{\parallel}$ and $V_S^{\perp}$ indicates whether we take the parallel or the perpendicular component of the G-register (equivalently, of the W-factor). We also abuse notation again and denote

$$V^{\parallel} \triangleq V^{\otimes s} \otimes W^{\parallel}, \qquad V^{\perp} \triangleq V^{\otimes s} \otimes W^{\perp}, \tag{11}$$

so that, with our notation,

$$X = V^{\otimes s} \otimes W = V^{\parallel} \oplus V^{\perp}.$$

We adopt the following convention. Let $S \subseteq [s]$, and let $x = (x_i)_{i \in S} \in \Lambda^{|S|}$ and $y = (y_i)_{i \in \overline{S}} \in \Lambda^{|\overline{S}|}$, where $\overline{S} = [s] \setminus S$. We write (x_S, y_{-S}) for the s-tuple $z = (z_1, \ldots, z_s)$ defined by

$$z_i \triangleq \begin{cases} x_i, & i \in S, \\ y_i, & i \notin S. \end{cases}$$

In the singleton case $S = \{i\}$, we write (x_i, y_{-i}) as shorthand. In particular, for a function $f : \Lambda^s \to \mathbb{R}$, the notation $f(x_S, y_{-S})$ denotes the evaluation of f at the point $(x_S, y_{-S}) \in \Lambda^s$. For notational convenience, we sometimes omit explicit subscripts: when the i-th coordinate is clear from context, we write $f(x, y_{-i})$ to mean $f(x_i, y_{-i})$.

In order to analyze the action of operators on these spaces, we need an operational characterization of the subspaces defined above. Focusing first on the H-component of X, given a function $f : \Lambda^{\otimes s} \to \mathbb{R}$ and a subset $S \subseteq [s]$, we would like to determine whether f lies in V_S. For this purpose, we have the following characterization.

Claim 4.1 (Characterization of V_S). *A function $f : \Lambda^{\otimes s} \to \mathbb{R}$ belongs to V_S if and only if* *the following two conditions hold:*

1. *f depends only on the coordinates in S (equivalently, for every fixing of $x \in \Lambda^{|S|}$ the function $g : \Lambda^{|\overline{S}|} \to \mathbb{R}$ defined by $g(y_{-S}) = f(x_S, y_{-S})$ is constant); and*

2. *For every $i \in S$ and every $y_{-i} \in \Lambda^{s-1}$, we have*

$$\mathbb{E}_{x_i \in \Lambda} \left[\, f(x_i, y_{-i}) \,\right] = 0,$$

where the expectation is uniform over the i-th coordinate.

Similarly, we have the corresponding decomposition for the G-register. To state it, let $P^{\|}, P^{\perp} : X \to X$ denote the (orthogonal) projections in the ambient space X onto the subspaces $V^{\|}$ and $V^{\perp}$, respectively, as defined in Equation (11). Also, for a fixed $g \in V(G)$, we define the function

$$f_g : \Lambda^{\otimes s} \to \mathbb{R}, \qquad f_g(x_1, \ldots, x_s) \triangleq f(x_1, \ldots, x_s, g).$$

We refer to the function f_g as the g-*cloud* of f.

Let $S \subseteq [s]$. For a function $f \in V_S^{\perp\!\!\!\!\perp}$, define

$$\overline{f} \triangleq P^{\|} f.$$

Note that

$$P^{\perp} f = f - \overline{f} \in V_S^{\perp}.$$

By definition, $\overline{f} \in V_S^{\|}$, and hence $\overline{f}$ is independent of the G-register. Accordingly, we slightly abuse notation and regard $\overline{f}$ as a function on Λ^s, defined by

$$\overline{f}(x) \triangleq \mathbb{E}_g[\, f_g(x) \,]\,,$$

where the expectation is uniform over $V(G)$.

4.1 The Linear Operators $\dot{G}$, $\widetilde{H}$ and Π

We now describe the linear operators acting on the space X defined in Equation (9) that are used in the s-wide product. Specifically, we consider the operators $\widetilde{H}, \dot{G}_1, \ldots, \dot{G}_s$ which induce walks over the vertices $V(G) \times V(H)$, and the noise operator Π.

Informally, the operator $\widetilde{H}$ corresponds to taking a local H-step within each g-cloud. Formally,

Definition 4.2. *The operator $\widetilde{H}$ is the linear operator on X defined as follows. For $f \in X$, $x \in \Lambda^s$ and $g \in V(G)$,*

$$(\widetilde{H} f)_g(x) = \mathbb{E}_{c \in C}[\, f_g(x + c) \,]\,,$$

where the expectation is uniform over the generating set C of H.

Definition 4.3. *For $i \in [s]$, we define the linear operator $\dot{G}_i$ on X as follows. For $f \in X$, $x = (x_1, \dots, x_s) \in \Lambda^s$ and $g \in V(G)$, let*

$$\mathrm{Rot}\,(g, x_i) = (h, \phi(x_i)).$$

Then

$$\big(\dot{G}_i f\big)_g(x) \;\triangleq\; f_h\big(\phi(x_i),\, x_{-i}\big).$$

In words: take a G-step from g according to the label x_i, updating the G-register and the i-th register, while keeping the other registers unchanged.

Last is Π which we refer to as the *noise operator* which acts on entire clouds.

Definition 4.4. *Let $\pi : V(G) \to \mathbb{R}$ be a function. Define the linear operator Π on X as follows. For $f \in X$, $x \in \Lambda^s$ and $g \in V(G)$,*

$$(\Pi f)_g(x) = \pi(g)\, f_g(x).$$

In words, Π multiplies each g-cloud by the scalar $\pi(g)$.

Throughout this section, we denote the *bias* of π by

$$\varepsilon \triangleq \left|\mathbb{E}_{g \in V(G)}\left[\pi(g)\right]\right| \tag{12}$$

As the notation suggests, we will typically assume that this bias is small, and certainly smaller than 1.

4.2 Action of $\widetilde{H}$

Lemma 4.5. *For every $S \subseteq [s]$ the following hold*

1. *$\widetilde{H}$ preserves V_S locally. That is,*

$$\widetilde{H} V_S^{\perp\!\!\perp} \subseteq V_S^{\perp\!\!\perp},$$

Moreover,

(a) *$\widetilde{H}$ acts trivially on $V_\emptyset^{\perp\!\!\perp}$.*

(b) *For $S \neq \emptyset$, for every $f \in V_S^{\perp\!\!\perp}$,*

$$\left\|\widetilde{H} f\right\| \leq \omega_H \left\|f\right\|. \tag{13}$$

2. $\widetilde{H}$ preserves $V^{\perp}$ and $V^{\parallel}$ globally. That is,

$$\widetilde{H}V^{\parallel} \subseteq V^{\parallel}, \qquad \widetilde{H}V^{\perp} \subseteq V^{\perp}.$$

Corollary 4.6. *For every $S \subseteq [s]$, $\widetilde{H}$ preserves $V_S^{\parallel}$ and $V_S^{\perp}$.*

We refer the reader to Figure 4 for an illustration of the action of $\widetilde{H}$.

Proof of Corollary 4.6. $\widetilde{H}$ preserves the spaces $V_S^{\perp\!\!\perp}$, $V^{\perp}$ and $V^{\parallel}$, so it preserves their intersections, and

$$V_S^{\perp} = V_S^{\perp\!\!\perp} \cap V^{\perp}, \qquad V_S^{\parallel} = V_S^{\perp\!\!\perp} \cap V^{\parallel}.$$

□

Proof of Lemma 4.5. We start with Item 1. Fix a function $f \in V_S^{\perp\!\!\perp}$ and recall that

$$\left(\widetilde{H}f\right)_g(x) = \mathbb{E}_{c\in C}\left[f_g\left(x+c\right)\right].$$

We omit the g in the subscript for ease of notations. According to the characterization in Claim 4.1, we need to prove two conditions. First, we need to verify that $\widetilde{H}f$ depends only on the coordinates in S. Indeed,

$$\begin{aligned}\widetilde{H}f\left(x_S, x_{\overline{S}}\right) &= \mathbb{E}_{c\in C}\left[f\left((x+c)_S, (x+c)_{\overline{S}}\right)\right]\\ &= \mathbb{E}_{c\in C}\left[f\left((x+c)_S, \cdot\right)\right]\\ &= (\widetilde{H}f)(x_S, \cdot).\end{aligned}$$

Second, we need to check that for every $i \in S$ and any fixing of y_{-i}, we get

$$\mathbb{E}_{x_i\in\Lambda}\left[\widetilde{H}f(x_i, y_{-i})\right] = 0.$$

Indeed,

$$\mathbb{E}_{x_i}\left[\widetilde{H}f(x_i, y_{-i})\right] = \mathbb{E}_{x_i}\left[\mathbb{E}_{c\in C}\left[f(x_i+c_i, y_{-i}+c_{-i})\right]\right] = \mathbb{E}_{c\in C}\left[\mathbb{E}_{x_i}\left[f(x_i+c_i, y_{-i}+c_{-i})\right]\right].$$

Changing the variable in the expectation we have that

$$\mathbb{E}_{c\in C}\left[\mathbb{E}_{x_i}\left[f(x_i+c_i, y_{-i}+c_{-i})\right]\right] = \mathbb{E}_{c\in C}\left[\mathbb{E}_{z_i}\left[f(z_i, y_{-i}+c_{-i})\right]\right] = 0,$$

where the last equality follows from the characterization in Claim 4.1 and the fact that $f \in V_S$.

Regarding parts (a) and (b) of Item 1, observe that $V_\emptyset \cong \mathbf{1}_H$, on which the random walk over the graph H acts trivially. For every $S \neq \emptyset$, $V_S \subseteq V_\emptyset^{\perp} \cong \mathbf{1}_H^{\perp}$ (here we really mean the duals of $V_\emptyset$ and $\mathbf{1}_H$, not the perpendicular part on the last register). On this subspace, H satisfies Equation (13). This proves Item 1.

For Item 2, assume $f \in V^\perp$ which recall is defined by $V^\perp \triangleq V^{\otimes s} \otimes W^\perp$ in Equation (11). Then,

$$\mathbb{E}_{g\in V(G)}\left[\left(\widetilde{H}f\right)_g(x)\right] = \mathbb{E}_{g\in V(G)}\left[\mathbb{E}_{c\in C}\left[f_g(x+c)\right]\right] = \mathbb{E}_{c\in C}\left[\mathbb{E}_{g\in V(G)}\left[f_g(x+c)\right]\right]$$

As $f \in V^\perp$,

$$\mathbb{E}_{g\in V(G)}\left[f_g(x+c)\right] = 0.$$

Hence, so is $\widetilde{H}f$.

Similarly, for $f \in V^\parallel = V^{\otimes s} \otimes W^\parallel$, we have that $f_g(x) = \mathbb{E}_{g'}\left[f_{g'}(x)\right] = \overline{f}(x)$, which does not depend on g. So

$$\left(\widetilde{H}f\right)_g = \mathbb{E}_{c\in C}\left[f_g(x+c)\right] = \mathbb{E}_{c\in C}\left[\overline{f}(x+c)\right],$$

which does not depend on g, which concludes the lemma. □

4.3 Action of Π

Lemma 4.7. *The operator Π satisfies the following properties:*

1. *For every $f \in V^\parallel$,*
$$\left\|P^\parallel \Pi f\right\| \le \varepsilon \left\|f\right\|.$$
In particular, when ε is small, Π maps $V^\parallel$ mostly into $V^\perp$.

2. *For every $S \subseteq [s]$, Π locally preserves V_S, i.e.,*
$$\Pi V_S^{\perp\!\!\!\perp} \subseteq V_S^{\perp\!\!\!\perp}.$$

We refer the reader to Figure 6 for an illustration of the action of Π.

Proof. For $f \in V^\parallel$ we have

$$\left\|P^\parallel \Pi f\right\| = \left\|P^\parallel \Pi P^\parallel f\right\| \le \left\|P^\parallel \Pi P^\parallel\right\| \cdot \left\|f\right\|,$$

so it suffices to show that $\left\|P^\parallel \Pi P^\parallel\right\| \le \varepsilon$. In tensor form, $P^\parallel = I^{\otimes s} \otimes J$, where J is the normalized all-ones matrix, and $\Pi = I^{\otimes s} \otimes \operatorname{diag}(\pi)$. Therefore,

$$P^\parallel \Pi P^\parallel = I^{\otimes s} \otimes \left(J \operatorname{diag}(\pi)\ J\right) = I^{\otimes s} \otimes \left(\mathbb{E}_g\left[\pi(g)\right]\ J\right) = \varepsilon \cdot I^{\otimes s} \otimes J,$$

which has operator norm ε, as desired.

The second property is immediate. For every S and $f \in V_S^{\perp\!\!\perp}$ we have, for each g,

$$(\Pi f)_g = \pi(g) \cdot f_g \in V_S.$$

□

4.4 Action of the $\dot{G}_i$-s

Informally, for each $i \in [s]$, the operator $\dot{G}_i$ may shift us along the i-th coordinate. It is therefore convenient to define the combined subspace

$$V_{S\pm i} \triangleq V_{S\setminus\{i\}} \oplus V_{S\cup\{i\}} \subseteq V^{\otimes s}.$$

Similarly to Claim 4.1, we provide an operative characterization of $V_{S\pm i}$

Claim 4.8. *A function f belongs to $V_{S\pm i}$ if and only if these two conditions hold:*

1. *f depends only on the coordinates in $S \cup \{i\}$; and*
2. *For every $j \in S \setminus \{i\}$, for each fixed y_{-j},*

$$\mathbb{E}_{x_j}[\, f(x_j, y_{-j})\,] = 0.$$

Proof of Claim 4.8. The forward direction follows immediately from Claim 4.1. For the converse, assume $f \in X$ satisfies both conditions of Claim 4.8, and decompose it as

$$f = f_1 + f_2, \qquad f_1(x_i, x_{-i}) \triangleq \mathbb{E}_{x_i'}[\, f(x_i', x_{-i})\,], \qquad f_2 \triangleq f - f_1.$$

Note that f_1 does not depend on x_i; combined with the two conditions of Claim 4.8, we conclude that f_1 satisfies the defining properties of $V_{S\setminus\{i\}}$. For f_2, observe that

$$\mathbb{E}_{x_i}[\, f_2(x_i, x_{-i})\,] = \mathbb{E}_{x_i}[\, f(x_i, x_{-i})\,] - \mathbb{E}_{x_i'}[\, f(x_i', x_{-i})\,] = 0.$$

Using the two conditions of Claim 4.8, together with the fact that the mean of f_2 over the i-th coordinate is zero, we conclude that f_2 satisfies the defining properties of $V_{S\cup\{i\}}$. Therefore, $f_2 \in V_{S\cup\{i\}}$, and consequently $f = f_1 + f_2 \in V_{S\pm i}$. □

We extend the subspace $V_{S\pm i}$ to subspaces of X.

$$V_{S\pm i}^{\perp\!\!\perp} \triangleq V_{S\pm i} \otimes W, \qquad V_{S\pm i}^{\perp} \triangleq V_{S\pm i} \otimes W^{\perp} \subseteq X.$$

Corollary 4.9. *A function $f \in X$ belongs to $V^{\perp\!\!\perp}_{S\pm i}$ if and only if for every $g \in V(G)$, $f_g \in V_{S\pm i}$.*

Lemma 4.10. *The operator $\dot{G}_i$ satisfies the following invariance properties:*

1. *For every $S \subseteq [s]$, $V^{\parallel}_S$ is $\dot{G}_i$ invariant; equivalently,*
$$\dot{G}_i V^{\parallel}_S \subseteq V^{\parallel}_S.$$
2. *For every $S \subseteq [s]$, $\dot{G}_i$ leaves $V^{\perp}_{S\pm i}$ invariant; equivalently,*
$$\dot{G}_i V^{\perp}_{S\pm i} \subseteq V^{\perp}_{S\pm i}.$$
3. *For every $S \subseteq [s]$ with $i \notin S$, $\dot{G}_i$ leaves only an ω_G-fraction of the norm in $V^{\perp}_S$. That is, for every $f \in V^{\perp}_S$,*
$$\left\| P^{\perp}_S \dot{G}_i f \right\| \leq \omega_G \left\| f \right\|.$$

We refer the reader to Figure 5 for an illustration of the action of the $\dot{G}_i$-s.

Proof of Lemma 4.10. For Item 1, fix $S \subseteq [s]$ and let $f \in V^{\parallel}_S$. Since $f \in V^{\parallel}$ is independent of g, the slice f_g is the same for all g; denote this common value by
$$\overline{f} \triangleq \mathbb{E}_g[f_g].$$

We first show that $\dot{G}_i f \in V^{\parallel}$, i.e., $(\dot{G}_i f)_g$ does not depend on g. For every x,
$$\begin{aligned}(\dot{G}_i f)_g(x) &= f_{\mathrm{Rot}(g,x_i)}\big(\phi(x_i), x_{-i}\big)\\ &= \overline{f}\big(\phi(x_i), x_{-i}\big),\end{aligned}$$

which is independent of g. This motivates the notation
$$\overline{\dot{G}_i f}(x) \triangleq \overline{f}\big(\phi(x_i), x_{-i}\big).$$

It remains to show that $\overline{\dot{G}_i f} \in V_S$. By Claim 4.1, it suffices to verify:

1. $\overline{\dot{G}_i f}$ depends only on coordinates indexed by S;
2. For every $j \in S$ and every fixed y_{-j}, we have $\mathbb{E}_{x_j}\left[\overline{\dot{G}_i f}(x_j, y_{-j})\right] = 0$.

The first condition is immediately inherited from $\overline{f}$. For the second condition, we treat the cases $j = i$ and $j \neq i$ separately.

If $j = i$, then

$$\mathbb{E}_{x_i\in\Lambda}\Big[\overline{\dot{G}_i f}(x_i, y_{-i})\Big] = \mathbb{E}_{x_i\in\Lambda}\Big[\overline{f}\big(\phi(x_i), x_{-i}\big)\Big] = \mathbb{E}_{x_i\in\Lambda}\Big[\overline{f}\big(x_i, x_{-i}\big)\Big] = 0,$$

where the second equality is because ϕ is a bijection of Λ and the third inequality follows the fact $\overline{f}$ has zero mean in coordinate i.

If $j \neq i$ with $j \in S$, then for the fixed $y'_{-j} = (\phi(x_i), x_{-\{i,j\}})$ we have

$$\mathbb{E}_{x_j\in\Lambda}\Big[\overline{\dot{G}_i f}(x_j, y_{-j})\Big] = \mathbb{E}_{x_j\in\Lambda}\Big[\overline{f}\big(\phi(x_i), x_j, x_{-\{i,j\}}\big)\Big] = \mathbb{E}_{x_j\in\Lambda}\Big[\overline{f}\big(x_j, y'_{-j}\big)\Big] = 0,$$

using the zero-mean property of $\overline{f}$ in coordinate j. This proves Item 1.

For Item 2, we begin by observing that $V^{\perp}_{S\pm i}$ is the intersection of $V^{\perp\!\!\perp}_{S\pm i}$ and $V^{\perp}$. Hence, it suffices to prove that each one of these subspaces is $\dot{G}_i$-invariant. Note that $\dot{G}_i$ is symmetric (self-adjoint). For a symmetric operator, invariance of a subspace implies invariance of its orthogonal complement. Since $V^{\|}$ is invariant by Item 1 and $(V^{\|})^{\perp} = V^{\perp}$ by definition, we conclude that $V^{\perp}$ is $\dot{G}_i$-invariant. To show that $V^{\perp\!\!\perp}_{S\pm i}$ is $\dot{G}_i$-invariant, let $f \in V^{\perp\!\!\perp}_{S\pm i}$, so that it satisfies the following two conditions:

1. For every $g \in V(G)$, the function f_g depends only on the coordinates in $S \cup \{i\}$;
2. For every $g \in V(G)$, every $j \in S \setminus \{i\}$, and every $y_{-j} \in \mathbb{R}^{s-1}$,

$$\mathbb{E}_{x_j}[\, f_g(x_j, y_{-j})\,] = 0.$$

We now verify that these properties also hold for $\dot{G}_i f$.

For the first condition, fix $j \notin S \cup \{i\}$. By the definition of $\dot{G}_i$,

$$(\dot{G}_i f)_g(x_i, x_j, x_{-\{i,j\}}) = f_{\mathrm{Rot}(g,x_i)}\big(\phi(x_i), x_j, x_{-\{i,j\}}\big),$$

which, by the assumption on f_g, does not depend on x_j. This establishes the first condition for $\dot{G}_i f$.

For the second condition, fix $j \in S \setminus \{i\}$ and $y_{-j} \in \mathbb{R}^{s-1}$. Then,

$$\mathbb{E}_{x_j}\Big[(\dot{G}_i f)_g(y_i, x_j, y_{-\{i,j\}})\Big] = \mathbb{E}_{x_j}\Big[f_{\mathrm{Rot}(g,y_i)}\big(\phi(y_i), x_j, y_{-\{i,j\}}\big)\Big] = 0,$$

where the last equality follows from the second condition on f. Hence, $\dot{G}_i f$ satisfies both defining properties of $V^{\perp\!\!\perp}_{S\pm i}$, and thus $\dot{G}_i$ leaves $V^{\perp\!\!\perp}_{S\pm i}$ invariant.

For Item 3, let $S \subseteq [s] \setminus \{i\}$ and let $f, f' \in V^{\perp}_{S}$. Our goal is to bound the inner product $\Big\langle f', \dot{G}_i f \Big\rangle$. During the proof, we reduce the inner product of functions on the wide product to inner products of functions on G and on H and it is convenient to normalize the quantities accordingly. We add a subscript to the inner products to help the reader keep track of the correct normalization.

$$\begin{aligned}\frac{\left\langle f', \dot{G}_i f\right\rangle_X}{|V(G)|\,|V(H)|} &= \mathbb{E}_{g\in V(G),\, x\in\Lambda^s}\left[\, f'_g(x_i, x_{-i})\, f_{\mathrm{Rot}(g,x_i)}\big(\phi(x_i), x_{-i}\big)\,\right] \\ &= \mathbb{E}_{x_{-i}}\Big[\mathbb{E}_{g\in V(G)}\big[\mathbb{E}_{x_i}\big[\, f'_g(x_i, x_{-i})\, f_{\mathrm{Rot}(g,x_i)}\big(\phi(x_i), x_{-i}\big)\,\big]\big]\Big].\end{aligned}$$

Since f and f' do not depend on x_i, the inner expectation satisfies

$$\mathbb{E}_{g\in V(G)}\big[\mathbb{E}_{x_i}\big[\, f'_g(x_i, x_{-i})\, f_{\mathrm{Rot}(g,x_i)}\big(\phi(x_i), x_{-i}\big)\,\big]\big] = \mathbb{E}_{g'\sim g}\left[\, f'_g(x)\, f_{g'}(x)\,\right]$$

where the expectation in the right hand side is over adjacent vertices g', g in G. Note that it does not depend on x_i, so we may return the expectation over it, so to obtain

$$\frac{\left\langle f', \dot{G}_i f\right\rangle_X}{|V(G)|\,|V(H)|} = \mathbb{E}_{x\in\Lambda^s}\left[\mathbb{E}_{g'\sim g}\left[\, f'_g(x)\, f_{g'}(x)\,\right]\right] \tag{14}$$

For each fixed x, consider the induced functions $f_x, f'_x : V(G) \to \mathbb{R}$ defined by $f_x(g) = f_g(x)$ and similarly for f'_x. Since $f, f' \in V_S^\perp \subseteq V^\perp$, they satisfy $\mathbb{E}_g[f_x(g)] = \mathbb{E}_g[f'_x(g)] = 0$. By the spectral expansion property of G,

$$\mathbb{E}_{g'\sim g}[\, f'_x(g) f_x(g')\,] = \frac{\langle f'_x, G f_x\rangle_G}{|V(G)|} \le \omega_G \frac{\|f'_x\|\;\|f_x\|}{|V(G)|}.$$

Plugging this back into Equation (14) and rearranging, we get

$$\frac{\left\langle f', \dot{G}_i f\right\rangle_X}{|V(H)|} \le \omega_G\, \mathbb{E}_{x\in\Lambda^s}[\, \|f'_x\|\;\|f_x\|\,] = \omega_G\, \frac{\langle \|f'_x\|\,, \|f_x\|\rangle_H}{|V(H)|},$$

where we think of $\|f_x\|$ and $\|f'_x\|$ as functions from Λ^s to $\mathbb{R}$ (via $x \to \|f_x\|$ and similarly for f'). Using Cauchy-Schwarz inequality,

$$\langle \|f'_x\|\,, \|f_x\|\rangle \le \|\|f'_x\|\| \cdot \|\|f_x\|\| = \|f'\| \cdot \|f\|\,.$$

We conclude that $\left\langle f', \dot{G}_i f\right\rangle_X \le \omega_G \|f'\| \cdot \|f\|$. Therefore,

$$\left\|P_S^\perp \dot{G}_i f\right\| \le \omega_G \|f\|\,,$$

concluding the proof of Item 3. □

We end this section with a figure that illustrates the action of all the operators we discussed in this section.

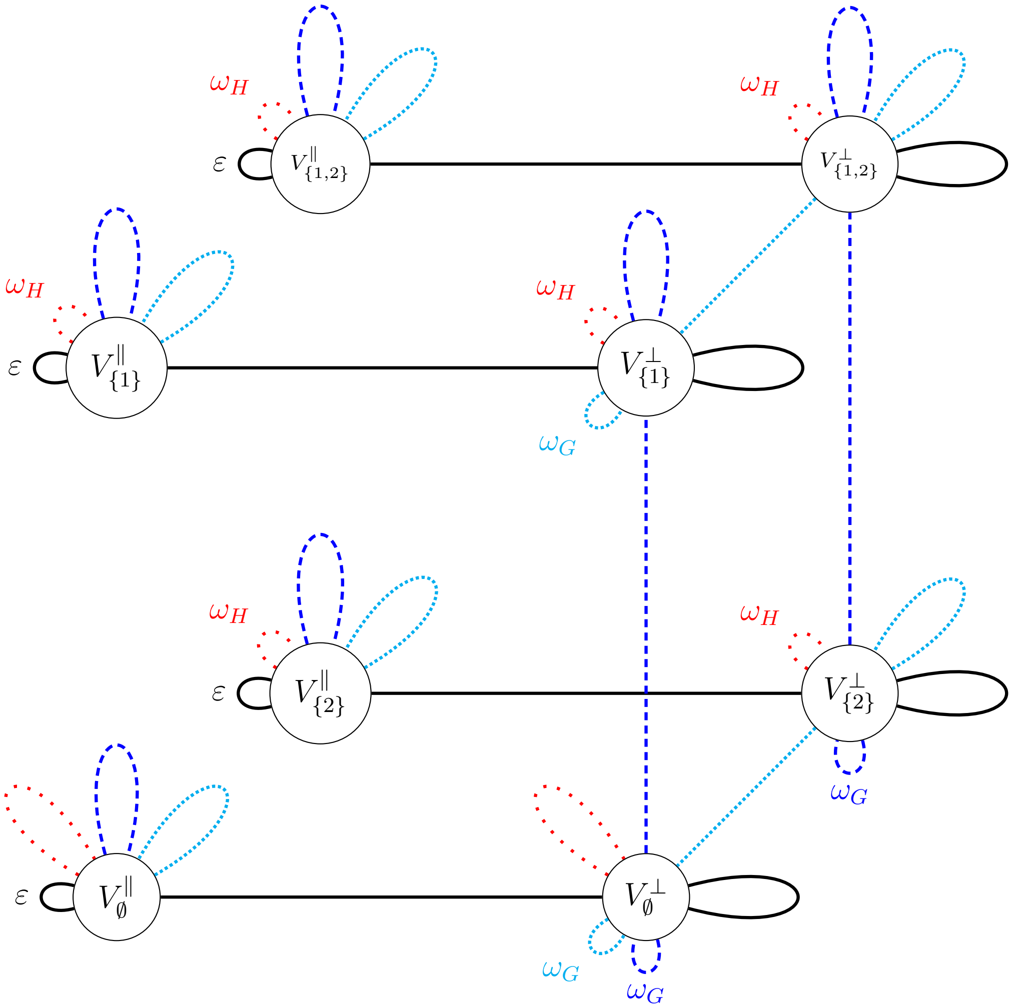


Figure 8: Illustration of the action of all operators: Π (black solid), $\widetilde{H}$ (red loosely dotted), $\dot{G}_1$ (blue dashed), and $\dot{G}_2$ (light-blue densely dotted).

4.5 Walks based on the wide product

Given $s \in \mathbb{N}$ and two graphs G and H with suitable parameters, we defined and analyzed the operators $\widetilde{H}$ and $\dot{G}_i$, for $i \in [s]$. These operators naturally refer to random walks on the set of vertices $V(G) \times V(H)$. Generally, we take a zig-zag walk. First an $\widetilde{H}$-step, then a $\dot{G}$-step, then an $\widetilde{H}$-step again. However, we still have the freedom to choose *which* $\dot{G}_i$ to use, as we have s distinct options in every step. For a word $W = W_{|W|} \cdots W_1$ over the alphabet $[s]$, we define the W-walk of H and G to be the graph with the random walk matrix

$$G \textcircled{\scriptsize w} H \triangleq \dot{G}_{W_{|W|}} \widetilde{H} \cdots \dot{G}_{W_1} \widetilde{H}$$

To construct small biased sets out of wide random walks, we identify the vertices of G with vectors from a small biased set, $S_{\texttt{outer}}$, so that each vertex $g \in V(G)$ correspond to a vector in $S_{\texttt{outer}}$. Similarly to the work of [TS17], the elements of our new biased set $S_{\texttt{new}}$ are defined to be sums of elements from $S_{\texttt{outer}}$, which correspond to the W-walks over the wide product of G and H. For each such a walk, we add a new vector to our new biased set $S_{\texttt{new}}$. The vector is the sum of the vectors from $S_{\texttt{outer}}$ that we encountered on the way. Explicitly, for a walk

$$(g_0, h_0) \to_{\widetilde{H}} (g_0, h_1) \to_{\dot{G}} (g_1, h_2) \to_{\widetilde{H}} \cdots \to_{\dot{G}} \left(g_{|W|}, h_{2|W|}\right),$$

we add the vector $\sum_{i=1}^{|W|} g_i$ to $S_{\texttt{new}}$. Given a linear test $\pi : \mathbb{Z}_2^k \to \{\pm 1\}$, the vertex g gets the value $\pi(g) = (-1)^{\langle g, \pi \rangle}$. This induces the operator Π.

Throughout, we will have a correspondence between words and linear operators. For a letter $\sigma \in [s]$, we define the corresponding operator that helps us track the bias as

$$\widehat{\sigma} = \Pi \dot{G}_\sigma \widetilde{H}.$$

We extend this definition to words. For a word $W = \sigma_{|W|} \cdots \sigma_1$,

$$\widehat{W} = \widehat{\sigma}_{|W|} \cdots \widehat{\sigma}_1,$$

As proved in [TS17], the bias of the new set on the test π can be expressed algebraically:

Claim 4.11. *For every word W over the alphabet $[s]$, the bias of the new set S_{new} is algebraically expressed as*

$$\mathsf{Bias}(S_{new}) = \mathbf{1}^t \Pi \dot{G}_{W_{|W|}} \widetilde{H} \cdots \Pi \dot{G}_{W_2} \widetilde{H} \Pi \dot{G}_{W_1} \widetilde{H} \mathbf{1} = \mathbf{1}^t \widehat{W} \mathbf{1}. \tag{15}$$

5 Conflict-Free Words

In this brief section, we present a recursive construction of the Gray–code–style string used in our register-reuse pattern.

Definition 5.1. *A nonempty word $w \in \Sigma^\star$* has a conflict *if every letter that appears in w appears at least twice (equivalently, no letter appears exactly once). A word w is* conflict-free *if there exists a letter $\sigma \in \Sigma$ that appears in w exactly once, or if it is the empty word.*

Definition 5.2. *A word $w \in \Sigma^\star$ is* double conflict-free *if either $|w| < 2$, or there exist two consecutive positions whose letters each appear exactly once in w.*

Definition 5.3. *A word $w \in \Sigma^\star$ is* (double) conflict-free on intervals *if every contiguous subword (interval) of w is (double) conflict-free.*

Lemma 5.4. *For every $s \in \mathbb{N}$ there is an explicit word w_s of length $2^s - 1$ over the alphabet $\{\sigma_1, \ldots, \sigma_s\}$ that is conflict-free on intervals.*

Proof. Define $w_1 = \sigma_1$ and recursively

$$w_s \;=\; w_{s-1}\, \sigma_s\, w_{s-1}.$$

We prove by induction on s that w_s is conflict-free on intervals. The base case is clear. For the inductive step, assume w_{s-1} is conflict-free on intervals, and let z be a contiguous subword of w_s. If z contains σ_s, then σ_s appears exactly once in z, so z is conflict-free. Otherwise, z is a contiguous subword of w_{s-1}, which is conflict-free by the induction hypothesis.

The lengths satisfy $|w_s| = 2\,|w_{s-1}| + 1$ with $|w_1| = 1$, yielding $|w_s| = 2^s - 1$. □

Fact 5.5. *The length $2^s - 1$ in Lemma 5.4 is optimal: every word of length 2^s over an alphabet of size s has a conflict.*

Lemma 5.6. *For every $s \in \mathbb{N}$ there is an explicit word w_{2s-1}, over the alphabet $\{\sigma_1, \ldots, \sigma_{2s-1}\}$, of length $3 \cdot 2^{s-1} - 2$ that is double conflict-free on intervals.*

Proof. As in Lemma 5.4, define $w_1 = \sigma_1$ and, for $s \geq 1$,

$$w_{2s+1} \;=\; w_{2s-1}\, \sigma_{2s}\, \sigma_{2s+1}\, w_{2s-1}.$$

We prove by induction on s that each w_{2s-1} is double conflict-free on intervals. The base case is immediate. For the inductive step, assume w_{2s-1} is double conflict-free on intervals and let z be a contiguous subword of w_{2s+1}. Exactly one of the following holds:

- z contains both σ_{2s} and σ_{2s+1}. Then these letters occur exactly once in z and are consecutive, so z is double conflict-free.
- z contains neither σ_{2s} nor σ_{2s+1}. Then z is a contiguous subword of w_{2s-1} and is double conflict-free by the induction hypothesis.
- z contains exactly one of σ_{2s} and σ_{2s+1}. W.l.o.g. assume z contains σ_{2s}. Write $z = z'\sigma_{2s}$ where z' is a suffix of w_{2s-1}. If $|z'| = 0$, then $|z| \leq 1$ and z is double conflict-free by definition. If $|z'| = 1$, say $z' = a$, then $z = a\,\sigma_{2s}$ has two consecutive letters, each appearing exactly once in z. If $|z'| \geq 2$, then by the induction hypothesis z' already contains two consecutive letters (none equal to σ_{2s}) that each appear exactly once in z', and this remains true after appending σ_{2s}. Thus z is double conflict-free.

Hence every contiguous subword of w_{2s+1} is double conflict-free. The lengths satisfy

$$|w_{2s+1}| \;=\; 2\,|w_{2s-1}| + 2, \qquad |w_1| = 1,$$

which solves to $|w_{2s-1}| = 3 \cdot 2^{s-1} - 2$. □

Fact 5.7. *The bound $3 \cdot 2^{s-1} - 2$ in Lemma 5.6 is tight: every word of length $3 \cdot 2^{s-1} - 1$ over an alphabet of size $2s - 1$ fails to be double conflict-free on intervals.*

6 Tracking the Norm

The following lemma exploits the key property of *double-conflict-free* words: every word contains a pair of consecutive letters that appear *exactly once* in the word. To analyze such a unique consecutive pair, we "zoom in" the corresponding pair of operators. Let σ and σ' such letters, and consider the product of their operators,

$$\widehat{\sigma}\widehat{\sigma}' = \Pi \dot{G}_\sigma \widetilde{H} \cdot \Pi \dot{G}_{\sigma'} \widetilde{H}.$$

Informally, because the letters σ and σ' appear only once, we can assume that before and after this block the process acts on subspaces that are independent of these coordinates. This observation motivates the definition of the space of functions that are independent of the registers σ and σ'.

$$V^{\perp\!\!\perp}_{-\{\sigma,\sigma'\}} = \bigoplus_{S:\sigma,\sigma'\notin S} V^{\perp\!\!\perp}_S, \tag{16}$$

and of the corresponding projection operator to this space, $P^{\perp\!\!\perp}_{-\{\sigma,\sigma'\}}$.

Lemma 6.1. *For every $\sigma, \sigma' \in [s]$,*

$$\left\| P^{\perp\!\!\perp}_{-\{\sigma,\sigma'\}} \dot{G}_\sigma \widetilde{H} \Pi \dot{G}_{\sigma'} P^{\perp\!\!\perp}_{-\{\sigma,\sigma'\}} \right\| \leq 2\omega_G + \varepsilon.$$

Proof. Denote

$$M \triangleq P^{\perp\!\!\perp}_{-\{\sigma,\sigma'\}} \dot{G}_\sigma \widetilde{H} \Pi \dot{G}_{\sigma'} P^{\perp\!\!\perp}_{-\{\sigma,\sigma'\}}.$$

We begin by showing that for every $S \subseteq [s]$, the operator M is $V^{\perp\!\!\perp}_S$-invariant. This is immediate for sets S that contain either σ or σ', since such components are annihilated by $P^{\perp\!\!\perp}_{-\{\sigma,\sigma'\}}$. For all other sets, observe that the space

$$V^{\perp\!\!\perp}_{S\pm\{\sigma,\sigma'\}} \triangleq V^{\perp\!\!\perp}_S \oplus V^{\perp\!\!\perp}_{S\cup\{\sigma\}} \oplus V^{\perp\!\!\perp}_{S\cup\{\sigma'\}} \oplus V^{\perp\!\!\perp}_{S\cup\{\sigma,\sigma'\}}$$

is invariant under each of the operators $\dot{G}_\sigma$, $\dot{G}_{\sigma'}$, $\widetilde{H}$, and Π. Moreover, the projection $P^{\perp\!\!\perp}_{-\{\sigma,\sigma'\}}$ annihilates all components except for the $V^{\perp\!\!\perp}_S$ part. Consequently, for any vector $v^{\perp\!\!\perp}_S \in V^{\perp\!\!\perp}_S$,

$$\dot{G}_\sigma \widetilde{H} \Pi \dot{G}_{\sigma'} P^{\perp\!\!\perp}_{-\{\sigma,\sigma'\}} v^{\perp\!\!\perp}_S \in V^{\perp\!\!\perp}_{S\pm\{\sigma,\sigma'\}},$$

and after applying the final projection, we remain within $V^{\perp\!\!\perp}_S$. Hence $MV^{\perp\!\!\perp}_S \subseteq V^{\perp\!\!\perp}_S$.

To prove Lemma 6.1, it thus suffices to bound the norm of the operator

$$M_S \triangleq P_S^{\perp\!\!\!\perp} \dot{G}_\sigma \widetilde{H} \Pi \dot{G}_{\sigma'} P_S^{\perp\!\!\!\perp}.$$

For every $S \subseteq [s] \setminus \{\sigma, \sigma'\}$, we analyze the action of M_S.

Informally, this problem can be reduced to the case $S = \emptyset$ and $(\sigma, \sigma') = (1, 2)$. This case is illustrated in Figure 8. The projection operators $P_\emptyset^{\perp\!\!\!\perp}$ forces us to start and terminate in $V_\emptyset^\perp$ or in $V_\emptyset^\parallel$. The operators in the middle, from right to left, correspond to taking a light blue-black-red-blue steps. The reader may observe that there is no way to do so without using at least one small self loop, of weight ε or ω_G.

Formally, fix $S \subseteq [s] \setminus \{\sigma, \sigma'\}$ and take a unit vector $v_S^{\perp\!\!\!\perp} \in V_S^{\perp\!\!\!\perp}$. Decompose it as $v_S^{\perp\!\!\!\perp} = v_S^\parallel + v_S^\perp$. Both components have norm at most 1. By the triangle inequality, it suffices to bound M_S on $V_S^\parallel$ and $V_S^\perp$ separately:

$$\left\| M_S v_S^{\perp\!\!\!\perp} \right\| \le \left\| M_S v_S^\perp \right\| + \left\| M_S v_S^\parallel \right\|. \tag{17}$$

By Item 1 in Lemma 4.10, the parallel component is preserved by $\dot{G}_{\sigma'}$. That is, although $\dot{G}_{\sigma'}$ may not act trivially on $V_S^\parallel$, it maps it back into the same space. Denote the resulting vector by $u_S^\parallel = \dot{G}_{\sigma'} v_S^\parallel \in V_S^\parallel$, so that

$$M_S v_S^\parallel = P_S \dot{G}_\sigma \widetilde{H} \Pi u_S^\parallel.$$

Since $\dot{G}_{\sigma'}$ has operator norm 1, we have $\left\| u_S^\parallel \right\| \le \left\| v_S^\parallel \right\| \le 1$. According to Lemma 4.7, the operator Π keeps only an ε-fraction on $V_S^\parallel$ and the rest flows into $V_S^\perp$:

$$\Pi u_S^\parallel = \varepsilon u_S^\parallel + u_S^\perp,$$

where $\left\| u_S^\perp \right\|, \left\| u_S^\parallel \right\| \le 1$, since Π itself has norm at most 1. Applying the triangle inequality once again, we get

$$\left\| M_S v_S^\parallel \right\| \le \varepsilon \left\| P_S \dot{G}_\sigma \widetilde{H} u_S^\parallel \right\| + \left\| P_S \dot{G}_\sigma \widetilde{H} u_S^\perp \right\| \le \varepsilon + \left\| P_S \dot{G}_\sigma \widetilde{H} u_S^\perp \right\|.$$

By Item 1 in Lemma 4.5, the operator $\widetilde{H}$ preserves the subspace $V_S^\perp$, so $\widetilde{H} u_S^\perp \in V_S^\perp$. Finally, by Item 3 in Lemma 4.10, the last term is bounded by ω_G. We therefore conclude that

$$\left\| M_S v_S^\parallel \right\| \le \varepsilon + \omega_G.$$

For the other term in Equation (17), again by Lemma 4.10, $\dot{G}_{\sigma'}$ keeps at most an ω_G-fraction of the norm in $V_S^\perp$ and the rest flows to $V_{S\cup\sigma'}^\perp$. That is, for some vectors $u_S^\perp, u_{S\cup\sigma'}^\perp$

of norms at most 1 in the corresponding subspaces, we have

$$\dot{G}_{\sigma'} v_S^{\perp} = \omega_G u_S^{\perp} + u_{S\cup\sigma'}^{\perp}.$$

Then, using the triangle inequality once again, we obtain

$$\left\| M_S v_S^{\perp} \right\| \le \omega_G \left\| P_S \dot{G}_\sigma \widetilde{H} \Pi u_S^{\perp} \right\| + \left\| P_S \dot{G}_\sigma \widetilde{H} \Pi u_{S\cup\{\sigma'\}}^{\perp} \right\| . \tag{18}$$

The first term is bounded by ω_G. For the second term, observe that the subspace $\bigoplus_{\sigma' \in K} V_K^{\perp\!\!\perp}$, which contains the space $V_{S\cup\{\sigma'\}}^{\perp}$ where $u_{S\cup\{\sigma'\}}^{\perp}$ resides, is invariant under all the operators P_S, $\dot{G}_\sigma$, $\widetilde{H}$, and Π. Moreover, P_S annihilates this entire subspace, so the second term in Equation (18) vanishes. Collecting the bounds and referring back to Equation (17), we conclude that

$$\|M_S\| \le \omega_G + \omega_G + \varepsilon = 2\omega_G + \varepsilon,$$

which concludes the proof. □

Theorem 6.2. *Assume that $2\omega_G + \varepsilon \le \frac{32}{20}\omega_H^5$. Let w be a word over the alphabet $[s]$, which is double-conflict-free on intervals. Then, the norm of its corresponding operator $W = \widehat{w}$ is bounded by*

$$\left\| \widehat{W} \right\| \le 10 \cdot (2\omega_H)^{|W|-2}$$

We prove Theorem 6.2, which concerns the norm of $\widehat{W}$, by establishing a broader statement that handles not only the norm but additional structural aspects of $\widehat{W}$, as described in the following lemma:

Lemma 6.3. *Under the assumptions of Theorem 6.2,*

1. *If we start at $V_\emptyset^{\perp}$, we lose at most one step. That is, for every vector $v_\emptyset^{\perp} \in V_\emptyset^{\perp}$,*

$$\left\| \widehat{W} v_\emptyset^{\perp} \right\| \le (2\omega_H)^{|W|-1}$$

2. *If we start from $V_\emptyset^{\perp}$ and end at $V_\emptyset^{\perp\!\!\perp}$, then we win every single step. That is, for every vector $v_\emptyset^{\perp} \in V_\emptyset^{\perp}$,*

$$\left\| P_\emptyset \widehat{W} v_\emptyset^{\perp} \right\| \le \frac{1}{2} (2\omega_H)^{|W|}$$

3. *In any case, we lose at most two steps. That is,*

$$\left\| \widehat{W} \right\| \le 10 \cdot (2\omega_H)^{|W|-2}$$

Proof. We prove by induction on $|W|$. The base case $|W| = 1$ is immediate for Item 1 and Item 3, since the bound exceeds 1. For Item 2, we expand $\widehat{W}$ as $\widehat{W} = \Pi \dot{G} \widetilde{H}$. By Item 2 in

Lemma 4.7, the operator Π preserves $V_\emptyset^{\perp\!\!\!\perp}$, and hence commutes with $P_\emptyset^{\perp\!\!\!\perp}$. Therefore, for every $v_\emptyset^\perp \in V_\emptyset^\perp$,

$$\left\| P_\emptyset^{\perp\!\!\!\perp} \Pi \dot{G} \widetilde{H} v_\emptyset^\perp \right\| \leq \|\Pi\| \left\| P_\emptyset^{\perp\!\!\!\perp} \dot{G} \widetilde{H} v_\emptyset^\perp \right\|.$$

We decompose the projection $P_\emptyset$ as $P_\emptyset = P_\emptyset^{\|} + P_\emptyset^{\perp}$, and obtain

$$\left\| P_\emptyset \Pi \dot{G} \widetilde{H} v_\emptyset^\perp \right\| \leq \left\| P_\emptyset^{\|} \dot{G} \widetilde{H} v_\emptyset^\perp \right\| + \left\| P_\emptyset^{\perp} \dot{G} \widetilde{H} v_\emptyset^\perp \right\|.$$

Since both $\widetilde{H}$ and $\dot{G}$ leave $V^\perp$ invariant, and $P_\emptyset^{\|}$ annihilates this space, the first term vanishes. By Item 1 in Lemma 4.5, the operator $\widetilde{H}$ preserves $V_\emptyset^\perp$, so that $\widetilde{H} v_\emptyset^\perp \in V_\emptyset^\perp$. Then, by Item 3 in Lemma 4.10, the second term is bounded by ω_G. Finally, using the assumption $2\omega_G + \varepsilon \leq \frac{32}{20}\omega_H^5$, we have $\omega_G \leq \omega_H$, completing the base case.

We now proceed to the inductive step. Let $W = W_k \cdots W_1$, and assume that the induction hypotheses hold for every word of length less than k that is double-conflict-free on intervals. We begin by proving Item 1.

$$\begin{aligned} \left\| \widehat{W} v_\emptyset^\perp \right\| &= \left\| \Pi \dot{G}_{w_k} \widetilde{H} \widehat{W}_{k-1} \cdots \widehat{W}_1 v_\emptyset^\perp \right\| \\ &\leq \|\Pi\| \left\| \dot{G}_{w_k} \right\| \left\| \widetilde{H} W_{k-1} \cdots W_1 v_\emptyset^\perp \right\| \\ &\leq \left\| \widetilde{H} W_{k-1} \cdots W_1 v_\emptyset^\perp \right\|. \end{aligned} \tag{19}$$

We distinguish between two cases, depending on the *last* step - whether or not it contributes an ω_H-factor through the final $\widetilde{H}$ operator. Denote $P_{-\emptyset}^{\perp\!\!\!\perp} = I - P_\emptyset^{\perp\!\!\!\perp}$, and substitute it into Equation (19):

$$\begin{aligned} \left\| \widetilde{H} W_{k-1} \cdots W_1 v_\emptyset^\perp \right\| &= \left\| \widetilde{H} \left(P_{-\emptyset}^{\perp\!\!\!\perp} + P_\emptyset^{\perp\!\!\!\perp} \right) W_{k-1} \cdots W_1 v_\emptyset^\perp \right\| \\ &\leq \left\| \widetilde{H} P_\emptyset^{\perp\!\!\!\perp} W_{k-1} \cdots W_1 v_\emptyset^\perp \right\| + \left\| \widetilde{H} P_{-\emptyset}^{\perp\!\!\!\perp} W_{k-1} \cdots W_1 v_\emptyset^\perp \right\| \\ &\leq \left\| \widetilde{H} \right\| \left\| P_\emptyset^{\perp\!\!\!\perp} W_{k-1} \cdots W_1 v_\emptyset^\perp \right\| + \left\| \widetilde{H} P_{-\emptyset}^{\perp\!\!\!\perp} \right\| \left\| \widehat{W}_{k-1} \cdots W_1 v_\emptyset^\perp \right\|. \end{aligned}$$

The first term is bounded by $\frac{1}{2}(2\omega_H)^{|W|-1}$ by the induction hypothesis for Item 2. For the second term, using Item 1 from the induction hypothesis together with part (b) of Item 1 in Lemma 4.5, we have

$$\left\| \widetilde{H} P_{-\emptyset}^{\perp\!\!\!\perp} \right\| \left\| \widehat{W}_{k-1} \cdots W_1 v_\emptyset^\perp \right\| \leq \omega_H \cdot (2\omega_H)^{|W|-2} = \tfrac{1}{2}(2\omega_H)^{|W|-1}.$$

Adding both contributions gives the desired bound:

$$\left\| \widehat{W} v_\emptyset^\perp \right\| \leq (2\omega_H)^{|W|-1}.$$

We now turn to Item 2, where we must show that for every vector $u_\emptyset \in V_\emptyset^{\perp\!\!\perp}$,

$$u_\emptyset^t \widehat{W} v_\emptyset^\perp \le \tfrac{1}{2}\omega_H^{|W|}.$$

Recall that in $W = W_{|W|} \cdots W_1$, there exists a pair of consecutive letters that appear exactly once. Denote these letters by σ and σ', and decompose

$$W = W_L\, \sigma\, \sigma'\, W_R, \qquad \widehat{W} = \widehat{W}_L\, \widehat{\sigma}\, \widehat{\sigma}'\, \widehat{W}_R = \widehat{W}_L\, \Pi \dot{G}_\sigma \widetilde{H} \Pi \dot{G}_{\sigma'} \widetilde{H} \widehat{W}_R.$$

Observe that the subspace $V_{-\{\sigma,\sigma'\}}^{\perp\!\!\perp}$ (defined in Equation (16)) is invariant under all operators Π, $\widetilde{H}$, and $\dot{G}_i$ (and their conjugates, as they are symmetric). For every $i \notin \{\sigma, \sigma'\}$, and thus under both $\widehat{W}_R$ and $\widehat{W}_L^t$. Moreover, $V_\emptyset^{\perp\!\!\perp} \subseteq V_{-\{\sigma,\sigma'\}}^{\perp\!\!\perp}$. Hence,

$$\widehat{W}_R v_\emptyset^\perp,\ \widehat{W}_L^t u_\emptyset \in V_{-\{\sigma,\sigma'\}}^{\perp\!\!\perp}.$$

Consequently, we may insert the projection $P_{-\{\sigma,\sigma'\}}^{\perp\!\!\perp}$ on both sides without changing the vectors. This allows us to rewrite

$$\begin{aligned} u_\emptyset^t \widehat{W}_L\, \widehat{\sigma}\, \widehat{\sigma}'\, \widehat{W}_R v_\emptyset^\perp &= u_\emptyset^t \widehat{W}_L P_{-\{\sigma,\sigma'\}}^{\perp\!\!\perp}\, \widehat{\sigma}\, \widehat{\sigma}'\, P_{-\{\sigma,\sigma'\}}^{\perp\!\!\perp} \widehat{W}_R v_\emptyset^\perp \\ &\le \left\| u_\emptyset^t \widehat{W}_L \right\| \left\| P_{-\{\sigma,\sigma'\}}^{\perp\!\!\perp}\, \widehat{\sigma}\, \widehat{\sigma}'\, P_{-\{\sigma,\sigma'\}}^{\perp\!\!\perp} \right\| \left\| \widehat{W}_R v_\emptyset^\perp \right\|. \end{aligned}$$

Applying Lemma 6.1 together with the induction hypotheses for Item 1 and Item 3, we obtain

$$\begin{aligned} 10\,(2\omega_H)^{|W_L|-2} \cdot (2\omega_G + \varepsilon) \cdot (2\omega_H)^{|W_R|-1} &\le 10\,(2\omega_H)^{|W_L|+|W_R|-3}\,(2\omega_G + \varepsilon) \\ &\le 10\,(2\omega_H)^{|W|-5}\,(2\omega_G + \varepsilon) \\ &\le \tfrac{1}{2}\,(2\omega_H)^{|W|}, \end{aligned}$$

where the final inequality follows from the assumption $2\omega_G + \varepsilon \le \frac{32}{20}\omega_H^5$. This completes the proof of Item 2.

Finally, we prove Item 3. We aim to bound $\left\|\widehat{W} v\right\|$ for a unit vector v. We divide the analysis into cases according to the *first* step, depending on whether we do or do not obtain an ω_H factor from the initial $\widetilde{H}$ step. Decompose

$$v = v_\emptyset^\parallel + v_\emptyset^\perp + v_{-\emptyset}^{\perp\!\!\perp},$$

where each component has norm at most 1. By the triangle inequality,

$$\left\|\widehat{W} v\right\| \le \left\|\widehat{W} v_\emptyset^\parallel\right\| + \left\|\widehat{W} v_\emptyset^\perp\right\| + \left\|\widehat{W} v_{-\emptyset}^{\perp\!\!\perp}\right\|. \tag{20}$$

The second term is bounded by $(2\omega_H)^{|W|-1}$, by Item 1 in the induction hypothesis, already established for words of length $|W|$.

For the third term, we gain an ω_H factor from the first $\widetilde{H}$ step:

$$\left\|\widehat{W}_1 v^{\perp\!\!\!\perp}_{-\emptyset}\right\| = \left\|\Pi\dot{G}\widetilde{H} v^{\perp\!\!\!\perp}_{-\emptyset}\right\| \le \|\Pi\| \left\|\dot{G}\right\| \left\|\widetilde{H} v^{\perp\!\!\!\perp}_{-\emptyset}\right\| \le \omega_H,$$

where the final inequality follows from part (b) of Item 1 in Lemma 4.5. Although we lose structural information about the resulting vector (since $\widehat{W}_1 v^{\perp\!\!\!\perp}_{-\emptyset}$ may or may not have a significant $V^{\perp\!\!\!\perp}_{\emptyset}$ component), the ω_H factor gained at the first step allows us to lose up to two additional steps, by Item 3 in the induction hypothesis:

$$\left\|\widehat{W} v^{\perp\!\!\!\perp}_{-\emptyset}\right\| \le \left\|\widehat{W}_{|W|}\cdots\widehat{W}_2\right\| \left\|\widehat{W}_1 v^{\perp\!\!\!\perp}_{-\emptyset}\right\| \le \omega_H \cdot 10 \cdot (2\omega_H)^{|W|-3} = 5(2\omega_H)^{|W|-2}.$$

For the first term in Equation (20), we lose the initial H step, since $\widetilde{H}$ acts trivially on $V^{\|}_{\emptyset}$. However, in this case $\widehat{W}_1 v^{\|}_{\emptyset}$ retains enough structure to guarantee that we lose at most one additional step. Indeed, since both $\widetilde{H}$ and $\dot{G}$ act trivially on $V^{\|}_{\emptyset}$, we have

$$\widehat{W}_1 v^{\|}_{\emptyset} = \Pi\dot{G}\widetilde{H} v^{\|}_{\emptyset} = \Pi v^{\|}_{\emptyset} = \varepsilon v^{\|}_{\emptyset} + v'^{\perp}_{\emptyset},$$

where $\left\|v'^{\perp}_{\emptyset}\right\| \le 1$. The first component is negligible, as $\varepsilon \le \omega_H^5$. The second component lies in $V^{\perp}_{\emptyset}$, on which we lose at most one step. Formally, using Item 1 and Item 3 from the induction hypotheses,

$$\left\|\widehat{W} v^{\|}_{\emptyset}\right\| \le \varepsilon \left\|\widehat{W}_{|W|}\cdots\widehat{W}_2 v^{\|}_{\emptyset}\right\| + \left\|\widehat{W}_{|W|}\cdots\widehat{W}_2 v'^{\perp}_{\emptyset}\right\| \le \varepsilon \cdot 10(2\omega_H)^{|W|-3} + (2\omega_H)^{|W|-2}.$$

Under the assumption $2\omega_G + \varepsilon \le \frac{32}{10}\omega_H^5$, this is bounded by $2(2\omega_H)^{|W|-2}$.

Combining all bounds and substituting back into Equation (20), we obtain

$$\left\|\widehat{W} v\right\| \le (2\omega_H)^{|W|-1} + 5(2\omega_H)^{|W|-2} + 2(2\omega_H)^{|W|-2} \le 10(2\omega_H)^{|W|-2},$$

which completes the inductive step for Item 3. □

From Theorem 6.2 and Lemma 6.3, it can be easily shown that the corresponding quadratic form $\mathbf{1}^T\widehat{W}\mathbf{1}$ is bounded by $(2\omega_H)^{|W|-O(1)}(2\omega_G + \varepsilon)$. Recall that We assumed $2\omega_G + \varepsilon = \omega_H^{\Omega(1)}$. If one slightly changes the constant in the Ω term, it can be shown that the quadratic form is effectively as if no H-step were lost,

$$\mathbf{1}^t\widehat{W}\mathbf{1} \le (2\omega_H)^{|W|}.$$

We do not elaborate on this refinement, as it does not affect our parameters in any significant way.

7 Proof of Theorem 1.1

In this section, we assemble the components and prove Theorem 1.1. We remark that, in terms of explicitness, our construction inherits the explicitness of Ta-Shma's construction: the only additional ingredient—the reuse sequence—can be generated efficiently.

Fix $0 < \delta < \frac{1}{2}$ and (throughout) take all logarithms to be base 2. Define

$$\begin{aligned} s &= \left\lceil \log\log\frac{1}{\delta} \right\rceil, \\ \omega_H &= 2^{-\sqrt{\log(1/\delta)}}, \\ \omega_G &= \frac{\omega_H^5}{5}, \\ \varepsilon_0 &= \omega_G^2, \\ d &= \left\lceil \omega_G^{-5/s} \right\rceil, \end{aligned}$$

Ingredients.

- Let $S \subseteq \{0,1\}^n$ be an ε_0-biased set of size $O(n/\varepsilon_0^4)$. Several explicit constructions are available (e.g., starting from [TS17]); the precise exponent will not matter for us.
- Let G be a d^s-regular graph on $|S|$ vertices with spectral expansion ω_G. Since d^s is a degree-5 polynomial in $1/\omega_G$, we may use expanders with comparatively loose parameters, such as those given by the zig–zag construction [RVW00]. Note also that, by our parameter choices, $|S| \geq \varepsilon_0^{-4} > d^s$.
- Let Λ be a group of size d, and let H be a Cayley graph on Λ^s with spectral expansion ω_H. Using small-bias sets [AGHP93], we obtain an explicit construction whose degree satisfies
$$c = O\left(\frac{(s\log d)^2}{\omega_H^2}\right).$$
Here the exponent 2 on ω_H is crucial.

Bias . By our choice of ω_G and ε_0, we may invoke Theorem 6.2 using the double conflict-free-on-intervals word of length ℓ which we may set to $2^{s/2}$ from Lemma 5.6. The resulting bias is

$$O\left((2\omega_H)^{\ell-2}\right) = O\left(2^{\ell} \cdot 2^{-(\ell-2)\sqrt{\log(1/\delta)}}\right).$$

As $\ell = 2^{s/2} = \sqrt{\log(1/\delta)}$, this becomes

$$\delta \cdot 2^{O\left(\sqrt{\log(1/\delta)}\right)}. \tag{21}$$

Set Size. By Theorem 6.2, the set size is

$$|V(G)| \cdot |V(H)| \cdot c^{\ell} = O\left(\frac{n}{\varepsilon_0^4} \cdot d^s \cdot c^{\ell}\right).$$

With $\varepsilon_0 = \omega_G^2$ and $d^s = \Theta(\omega_G^{-5})$, the prefactor is $n \cdot 2^{O(\sqrt{\log(1/\delta)})}$. Moreover,

$$c^{\ell} = \left(\frac{s \log d}{\omega_H}\right)^{2\ell} = (s \log d)^{2\ell} \cdot \omega_H^{-2\ell}.$$

Since $s \log d = \log(d^s) = \log(\omega_G^{-5}) = \Theta(\sqrt{\log(1/\delta)})$, we have

$$(s \log d)^{2\ell} = 2^{O\left(\sqrt{\log(1/\delta)} \cdot \log\log(1/\delta)\right)} \quad \text{and} \quad \omega_H^{-2\ell} = \left(2^{\sqrt{\log(1/\delta)}}\right)^{2\ell} = \frac{1}{\delta^2}.$$

Putting everything together, we get size

$$\frac{n}{\delta^2} \cdot 2^{O\left(\sqrt{\log(1/\delta)} \cdot \log\log(1/\delta)\right)}. \tag{22}$$

Normalizing the bias to ε. Recall Equation (21) which gives a bound of $\delta \cdot 2^{\Theta\left(\sqrt{\log(1/\delta)}\right)}$ on the bias. To make the resulting bias bounded by ε, we rescale δ as a function of ε by setting

$$\delta = \varepsilon \cdot 2^{-B\sqrt{\log(1/\varepsilon)}}$$

for a sufficiently large absolute constant $B > 0$ (large enough to dominate the hidden constant in the $\Theta(\cdot)$ appearing in Equation (21)). With this choice, the bias is bounded by ε and we move to analyze the resulting set size in terms of ε. Substituting δ to Equation (22), observing that

$$\sqrt{\log(1/\delta)} = \sqrt{\log(1/\varepsilon) + B\sqrt{\log(1/\varepsilon)}} = \sqrt{\log(1/\varepsilon)} + O(1),$$

yields the desired bound on the set size

$$2^{O\left(\sqrt{\log(1/\delta)}\ \log\log(1/\delta)\right)} = 2^{O\left(\sqrt{\log(1/\varepsilon)}\ \log\log(1/\varepsilon)\right)}.$$

This completes the proof.

References

[AGHP93] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Addendum to: "Simple constructions of almost k-wise independent random variables". *Random Structures*

Algorithms, 4(1):119–120, 1993.

[BATS11] Avraham Ben-Aroya and Amnon Ta-Shma. A combinatorial construction of almost-Ramanujan graphs using the zig-zag product. *SIAM J. Comput.*, 40(2):267–290, 2011.

[BATS13] Avraham Ben-Aroya and Amnon Ta-Shma. Constructing small-bias sets from algebraic-geometric codes. *Theory Comput.*, 9:253–272, 2013.

[BD22] Guy Blanc and Dean Doron. New near-linear time decodable codes closer to the GV bound. In *37th Computational Complexity Conference*, volume 234 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 10, 40. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2022.

[CHHL19] Eshan Chattopadhyay, Pooya Hatami, Kaave Hosseini, and Shachar Lovett. Pseudorandom generators from polarizing random walks. *Theory Comput.*, 15:Paper No. 10, 26, 2019.

[CM23] Gil Cohen and Gal Maor. Random walks on rotating expanders. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 971–984, 2023.

[CRS12] Gil Cohen, Ran Raz, and Gil Segev. Non-malleable extractors with short seeds and applications to privacy amplification. In *2012 IEEE 27th Conference on Computational Complexity—CCC 2012*, pages 298–308. IEEE Computer Soc., Los Alamitos, CA, 2012.

[FK18] Michael A. Forbes and Zander Kelley. Pseudorandom generators for read-once branching programs, in any order. In *59th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2018*, pages 946–955. IEEE Computer Soc., Los Alamitos, CA, 2018.

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

[JMRW25] Fernando Granha Jeronimo, Tushant Mittal, Sourya Roy, and Avi Wigderson. Almost-Ramanujan Expanders From Arbitrary Expanders via Operator Amplification. *SIAM J. Comput.*, 54(5):FOCS22–120–FOCS22–158, 2025.

[JQST20] Fernando Granha Jeronimo, Dylan Quintana, Shashank Srivastava, and Madhur Tulsiani. Unique decoding of explicit ε-balanced codes near the Gilbert-Varshamov bound. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science*, pages 434–445. IEEE Computer Soc., Los Alamitos, CA, [2020] ©2020.

[JST21] Fernando Granha Jeronimo, Shashank Srivastava, and Madhur Tulsiani. Near-linear time decoding of Ta-Shma's codes via splittable regularity. In *STOC '21—Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1527–1536. ACM, New York, [2021] ©2021.

[MRRW77] Robert J. McEliece, Eugene R. Rodemich, Jr. Rumsey, Howard, and Lloyd R. Welch. New upper bounds on the rate of a code via the delsarte–macwilliams inequalities. *IEEE Transactions on Information Theory*, 23(2):157–166, March 1977.

[NN93] Joseph Naor and Moni Naor. Small-bias probability spaces: efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993.

[Raz05] Ran Raz. Extractors with weak random seeds. In *STOC'05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 11–20. ACM, New York, 2005.

[RV05] Eyal Rozenman and Salil Vadhan. Derandomized squaring of graphs. In *Approximation, randomization and combinatorial optimization*, volume 3624 of *Lecture Notes in Comput. Sci.*, pages 436–447. Springer, Berlin, 2005.

[RVW00] Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 3–13. IEEE, 2000.

[TS17] Amnon Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 238–251, 2017.

[Var57] R. R. Varshamov. Estimate of the number of signals in error-correcting codes. *Doklady Akademii Nauk SSSR*, 117:739–741, 1957. English translation reprinted in I. F. Blake (ed.), *Algebraic Coding Theory: History and Development*, Dowden, Hutchinson & Ross, 1973, pp. 68–71.

[Vio09] Emanuele Viola. The sum of d small-bias generators fools polynomials of degree d. *Comput. Complexity*, 18(2):209–217, 2009.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il