Electronic Colloquium on Computational Complexity, Report No. 189 (2026)

Marton's conjecture in polynomial time

Srinivasan Arunachalam [*]
IBM Research

Arkopal Dutt [†]
IBM Research

Sabee Grewal [‡]
IBM Research
Columbia University

Aparna Gupte [§]
MIT

Abstract

Gowers, Green, Manners, and Tao (Annals '25) recently resolved Marton's polynomial Freiman–Ruzsa conjecture. We give an algorithmic counterpart to their result: given uniform sampling and membership-oracle access to a set $A \subseteq \mathbb{F}_2^n$ with doubling constant at most K, our algorithm outputs a subspace of size at most $|A|$ whose $K^{O(1)}$ translates cover A. The algorithm runs in $\mathrm{poly}(n, K)$ time. As applications, we obtain polynomial-time algorithms for a variety of learning problems, including quadratic Goldreich–Levin, improper agnostic tomography of stabilizer states, and tomography of quantum states with bounded stabilizer extent.

[*] srinivasan.arunachalam@ibm.com
[†] arkopal@ibm.com
[‡] sabee@ibm.com
[§] agupte@mit.edu. This work was done while the author was an intern at IBM Research.

ISSN 1433-8092

Contents

1 Introduction

In a recent breakthrough, Gowers, Green, Manners, and Tao [GGMT25, GGMT24] proved Marton's conjecture, also known as the polynomial Freiman–Ruzsa (PFR) conjecture, first over $\mathbb{F}_2^n$ and subsequently over all abelian groups of bounded torsion. The theorem characterizes, up to polynomial losses, the algebraic structure of sets A with small *doubling constant*. For a nonempty set $A \subseteq \mathbb{F}_2^n$, define its *doubling constant* as

$$K(A) := \frac{|A + A|}{|A|} = \frac{|\{x + y : x, y \in A\}|}{\{x : x \in A\}}.$$

If A is a subspace, then $K(A) = 1$. More generally, PFR[1] asserts that a set with small doubling constant must resemble a subspace. Over characteristic 2, which is the version relevant to this work, the theorem is as follows:

Theorem 1.1 (Polynomial Freiman–Ruzsa (PFR) theorem [GGMT25])**.** *There is a polynomial $P\colon \mathbb{R}_+ \to \mathbb{R}_+$ such that, for every $n \geq 1$, $K \geq 1$, and nonempty set $A \subseteq \mathbb{F}_2^n$ satisfying $|A + A| \leq K|A|$, there is a subspace $V \leq \mathbb{F}_2^n$ with $|V| \leq |A|$ such that A is covered by at most $P(K)$ translates of V.*

The theorem formalizes the principle that small additive growth forces algebraic structure. If $K(A)$ is small, then A cannot behave like an arbitrary subset of $\mathbb{F}_2^n$; rather, it must be covered by only $K^{O(1)}$ translates of a subspace V whose size is at most $|A|$. We refer to such a subspace as a *PFR subspace* for A. Thus, sets with small doubling exhibit subspace-like structure, with the extremal example being a subspace itself.

PFR and related additive-combinatorial structure theorems have played an important role throughout theoretical computer science, with applications to property testing [Sam07], pseudorandomness [ZB11], communication complexity [BLR14], coding theory [BDL13, ADL14], worst-case-to-average-case reductions [AGGS22, AGG+24], sparsification [BNOZ25], and quantum computing [AD25, AD26b, BvDH25, MT25, HBvD+26, AD26a]. In these applications, PFR is used as a structural theorem: it guarantees the existence of a subspace with the desired additive structure.

From an algorithmic perspective, the PFR theorem leaves open a natural question. The theorem guarantees that a small-doubling set is governed by a low-complexity algebraic object—a single subspace and polynomially many of its translates—but the statement is purely *existential*. In the settings we consider, the set $A \subseteq \mathbb{F}_2^n$ may be exponentially large and therefore cannot be given explicitly. We instead assume two natural forms of oracle access: uniform samples from A and a membership oracle that, on input $x \in \mathbb{F}_2^n$, determines whether $x \in A$. Such implicit access models are standard through theoretical computer science when the underlying object is too large to enumerate: sampling reveals typical elements of A, while membership queries allow the algorithm to probe points of its choice. Thus, the model gives succinct access to A without assuming an explicit representation of the set. This leads to the algorithmic analogue of the PFR problem:

Given uniform sample access as well as membership oracle access to a set $A \subseteq \mathbb{F}_2^n$ with doubling constant at most K, can we find a PFR subspace for A in time $\mathrm{poly}(n, K)$*?*

1.1 Main result

Our main result answers this question in the affirmative.

[1] Ruzsa's bounded-torsion analog of Freiman's theorem and subsequence improvements [Ruz99, GT09] already showed that a set with doubling constant K is contained in a single subspace whose size is exponential in K times $|A|$. Marton [Ruz99] conjectured a substantially sharper description: rather than embedding A into one exponentially larger subspace, one should be able to cover A by *polynomially* many translates of a subspace no larger than A itself. This became known as the *polynomial Freiman–Ruzsa conjecture* and was resolved, nearly three decades later, by Gowers, Green, Manners, and Tao [GGMT25].

Theorem 1.2 (Algorithmic PFR). *Let $0 < \delta < 1$, and let $A \subseteq \mathbb{F}_2^n$ be nonempty and satisfy $|A + A| \leq K|A|$. Given uniform sampling and membership-oracle access to A, there is a randomized algorithm that, with probability at least $1 - \delta$, outputs a basis for a subspace $V \leq \mathbb{F}_2^n$ such that $|V| \leq |A|$ and A can be covered by at most $K^{O(1)}$ translates of V. The algorithm uses $\mathrm{poly}(n, K, \log(1/\delta))$ time, samples, and membership queries.*

Prior work. The algorithmic study of PFR was initiated recently in [ACDG26, CBA+26], motivated in part by its role in quantum learning [AD25, AD26a]. These works gave classical and quantum algorithms for recovering PFR structure with running time $\mathrm{poly}(n, 2^K)$. The underlying ideas arose through a somewhat indirect route: from a quantum algorithm for agnostically learning stabilizer states [CGYZ25], through its subsequent dequantization [BC26], and ultimately to classical algorithms for recovering additive structure [ACDG26]. In additive combinatorics, often K is viewed as being "small" (say constant or logarithmic) in terms of the ambient dimension, in which case their algorithm is efficient. The exponential dependence on K, however, is a significant limitation when the doubling constant grows with the problem size; which happens to be the case for several applications of interest, including those considered here, K may itself be polynomially related to the underlying parameters. This work improves the dependence on K from exponential to polynomial.

Our approach. Conceptually, our approach differs substantially from the previous algorithmic PFR results. Rather than reaching PFR through quantum learning-theoretic machinery, we return directly to the GGMT proof and show how to turn its entropic argument into an efficient algorithm. Several new ideas are necessary for algorithmizing the existential proof. The GGMT proof follows an iterative argument that starts out with the uniform distribution over the set A, and repeatedly transforms the distribution in order to eventually be close to the uniform distribution on a PFR subspace for A. This is done by making choices and comparing quantities appear difficult to compute efficiently, and the intermediate distributions may have exponentially large support and therefore cannot be held explicitly. Moreover, even if these distributions can be represented implicitly, one must ensure that the cost of accessing them does not grow prohibitively over the course of the recursion.

The main technical contribution of this work is to overcome these obstacles while preserving polynomial dependence on both n and K. Section 2 gives a self-contained overview of the GGMT argument and explains how each ingredient is implemented algorithmically.

1.2 Applications

Beyond giving an efficient algorithmic form of PFR, our result can be used as a general-purpose primitive for recovering additive structure inside larger algorithms. We illustrate this through two groups of applications. First, on the classical side, we obtain a polynomial-time quadratic Goldreich–Levin algorithm and, consequently, related learning problems. Second, we prove a robust variant of algorithmic PFR and use it to obtain polynomial-time algorithms for several problems in quantum learning, including agnostic tomography of stabilizer states, estimating stabilizer fidelity, and tomography of states with bounded stabilizer extent.

1.2.1 Quadratic Goldreich-Levin

A fundamental algorithmic primitive in Fourier analysis is the *Goldreich-Levin* algorithm [GL89]. Given query access to a bounded function $f : \mathbb{F}_2^n \to [-1, 1]$ and a threshold $\tau > 0$, the algorithm efficiently finds all linear characters $x \mapsto (-1)^{\langle \alpha, x \rangle}$ having correlation at least τ with f. Equivalently, it finds the large Fourier coefficients of f in time polynomial in n and $1/\tau$. Goldreich–Levin and its variants have become basic tools in coding theory [GL89, Tre04], computational learning theory [KM93], pseudorandomness and cryptography [GL89, HILL99], and complexity theory.

There is a useful higher-order perspective on this result. The U^2 Gowers norm measures linear structure, and its inverse theorem states that

$$\|f\|_{U^2} \text{ large} \qquad \Longrightarrow \qquad \left|\mathop{\mathbb{E}}_x f(x)(-1)^{\langle \alpha, x\rangle}\right| \text{ large}$$

for some $\alpha \in \mathbb{F}_2^n$. From this viewpoint, Goldreich–Levin is an *algorithmic inverse theorem for the U^2 norm*: it not only guarantees the existence of a correlated linear phase, but efficiently finds one.

The next level of this hierarchy is the U^3 norm, whose associated structured objects are quadratic phases. In a seminal work, Samorodnitsky [Sam07] proved, assuming the PFR conjecture, an inverse theorem of the form

$$\|f\|_{U^3} \text{ large} \qquad \Longrightarrow \qquad \left|\mathop{\mathbb{E}}_x f(x)(-1)^{q(x)}\right| \text{ large}$$

for some quadratic polynomial $q : \mathbb{F}_2^n \to \mathbb{F}_2$. The resolution of PFR therefore gives an unconditional polynomial inverse theorem for the U^3 norm.

This raises the corresponding algorithmic question: can one efficiently find the correlated quadratic phase promised by the inverse theorem? Tulsiani and Wolf [TW14] initiated the study of this *quadratic Goldreich–Levin problem* and gave an algorithm using $O(n^4)$ queries, but with running time 2^n. More recently, Briët and Castro [BC26] improved the query complexity to $O(n^2)$ and obtained $n^{O(\log n)}$ running time. Using our algorithmic PFR theorem, we obtain polynomial dependence on both parameters.[2]

Theorem 1.3 (Quadratic Goldreich–Levin)**.** *Let $f : \mathbb{F}_2^n \to [-1,1]$, let $\varepsilon, \delta > 0$. Let*

$$\tau \coloneqq \max_{\substack{q:\mathbb{F}_2^n \to \mathbb{F}_2 \\ q \text{ quadratic}}} \left|\mathop{\mathbb{E}}_{x\in\mathbb{F}_2^n}[f(x)(-1)^{q(x)}]\right|$$

There is an algorithm that, given query access to f, runs in time

$$\mathrm{poly}(n, 1/\tau, \log 1/\delta)$$

and, with probability at least $1-\delta$, outputs a quadratic polynomial $q : \mathbb{F}_2^n \to \mathbb{F}_2$ satisfying

$$\left|\mathop{\mathbb{E}}_{x\in\mathbb{F}_2^n}[f(x)(-1)^{q(x)}]\right| \geq \tau^C,$$

for some universal constant $C > 1$.

As in the work of Tulsiani and Wolf, a quadratic Goldreich–Levin algorithm also yields an efficient quadratic decomposition theorem: a bounded function can be decomposed into a polynomial number of quadratic phases together with error terms that are small in U^3 and L_1.

Corollary 1.4. *Let $\varepsilon \geq 0$ and $f : \mathbb{F}_2^n \to [-1,1]$. Given query access to f, there is a $\mathrm{poly}(n, 1/\varepsilon)$-time algorithm that outputs with probability $\geq 2/3$ a decomposition*

$$f = c_1(-1)^{p_1(\cdot)} + \cdots + c_r(-1)^{p_r(\cdot)} + g + h,$$

where the c_i are constants, p_is are quadratic polynomials, $r \leq \mathrm{poly}(1/\varepsilon)$, $\|g\|_{U^3} \leq \varepsilon$ and $\|h\|_1 \leq \varepsilon$.

Furthermore, Theorem 1.2 also has other learning-theoretic consequences. Since quadratic polynomials over $\mathbb{F}_2$ are precisely the codewords of the second-order Reed–Muller code $\mathrm{RM}(n,2)$, it gives a polynomial-time improper agnostic learner for $\mathrm{RM}(n,2)$.[3] In addition, standard reductions from agnostic learning [KSS92, Fel09] yield polynomial-time PAC learners for low-weight linear thresholds of quadratic phases.

[2] The term *quadratic Goldreich–Levin* is used in the literature for several closely related guarantees. Tulsiani and Wolf [TW14] study a multiplicative approximation to the maximum quadratic correlation, whereas Briët and Castro [BC26] obtain an additive approximation. Our result is in the former regime.

[3] An open question is whether one can obtain a *proper* agnostic learner for $\mathrm{RM}(n,2)$, as well as for stabilizer states.

1.2.2 Quantum applications

The study of learning near-stabilizer structure has developed through a sequence of works, including stabilizer learning and estimation [Mon17, GNW21, GIKL24], tomography of states with high stabilizer dimension [GIKL25, LOH24], and agnostic tomography of stabilizer states [GIKL26, CGYZ25].

More recently, Arunachalam and Dutt [AD25, AD26a, AD26b] uncovered a surprising connection between these quantum learning problems and additive combinatorics, showing that efficient algorithmic PFR would imply efficient algorithms for several of them. Their reductions, however, require a slightly more robust form of algorithmic PFR than the one stated in Theorem 1.2: instead of receiving perfectly uniform samples from the underlying small-doubling set, the algorithm must tolerate samples drawn from a distribution that is only approximately uniform. We show that our algorithm extends to precisely this setting, and this allows us to make the resulting quantum algorithms fully polynomial-time.

Approximate algorithmic PFR theorem. Recall that Theorem 1.2 assumes uniform sampling and membership-oracle access to a small-doubling set A. For the quantum applications, the sampling distribution arising from the reduction need not be exactly uniform. We therefore consider a distribution μ_A supported on A that is R-uniform, meaning that

$$\frac{1}{R|A|} \leq \mu_A(x) \leq \frac{R}{|A|}$$

for every $x \in A$. We prove that algorithmic PFR remains efficient under this weaker sampling model: given membership-oracle access to A and samples from an R-uniform distribution on A, one can recover a PFR subspace in poly $\left(n, K, R, \log \frac{1}{\delta}\right)$ time.

The main idea is to *uniformize* the samples before invoking our PFR algorithm. We define a lazy random walk on A whose stationary distribution is uniform on A. The upper bound in the R-uniformity condition gives a warm start for this walk, while the lower bound, together with the small-doubling hypothesis, yields a polynomial lower bound on its conductance. A warm-start mixing bound of Lovász and Simonovits then implies that the walk reaches a distribution arbitrarily close to uniform in $\mathrm{poly}(K, R, 1/\varepsilon)$ steps. Replacing each uniform sample required by Theorem 1.2 with an independently uniformized sample gives the desired robust version of algorithmic PFR.

Consequences for quantum learning. We now combine approximate algorithmic PFR with the reductions developed in [AD26a, AD26b]. For a pure state $|\psi\rangle$, define its *stabilizer fidelity* by

$$\mathcal{F}_{\mathcal{S}}(|\psi\rangle) = \max_{|\varphi\rangle \in \mathrm{Stab}} \{|\langle\varphi|\psi\rangle|^2\},$$

where Stab is the class of n-qubit stabilizer states. We obtain the following consequences.

1. **Self-correction.** Given copies of a pure state $|\psi\rangle$ satisfying $\mathcal{F}_{\mathcal{S}}(|\psi\rangle) \geq \tau$, there is a $\mathrm{poly}(n, 1/\tau)$-time algorithm that outputs a stabilizer state $|\varphi\rangle$ satisfying

   $$|\langle\varphi|\psi\rangle|^2 \geq \tau^C$$

   for some universal constant $C > 1$.

2. **Improper agnostic tomography of stabilizer states.** Combining self-correction with the boosting framework of [AD26b] gives a polynomial-time improper agnostic tomography algorithm for stabilizer states. Given copies of an arbitrary pure state $|\psi\rangle$, the algorithm outputs an efficiently representable state $|\widehat{\psi}\rangle$ satisfying

   $$|\langle\widehat{\psi}|\psi\rangle|^2 \geq \mathcal{F}_{\mathcal{S}}(|\psi\rangle) - \varepsilon$$

   in time polynomial in n and $1/\varepsilon$.

3. **Tomography of bounded-stabilizer-extent states.** Combining self-correction with the extent-boosting framework of [AD26b] gives a tomography algorithm for pure states of bounded stabilizer extent. If $|\psi\rangle$ has stabilizer extent at most ξ, then it can be learned to trace-distance error ε using $\mathrm{poly}(n, \xi, 1/\varepsilon)$ time and copies.

4. **Improper agnostic tomography of high stabilizer-dimension states.** Prior work [AD26a] showed that an efficient algorithmic PFR theorem implies self-correction for states of high stabilizer dimension.[4]. Concretely, if $|\psi\rangle$ has fidelity at least τ with a state of stabilizer dimension $n - t$, then in time $\mathrm{poly}(n, 2^t, 1/\tau)$ we can find a stabilizer state $|\varphi\rangle$ satisfying

$$|\langle\varphi|\psi\rangle|^2 \geq \mathrm{poly}(2^{-t}\tau).$$

   Combined with the boosting framework of [AD26b], this gives a $\mathrm{poly}(n, 2^t, 1/\varepsilon)$-time improper agnostic tomography algorithm for states with stabilizer dimension $n-t$. This extends the polynomial-time algorithm of [GIKL25] to the agnostic setting, improving on the previously known quasipolynomial-time algorithm even for $t = O(\log n)$ [CGYZ25].

5. **Learning Clifford structure.** The reduction of [DJJ+26] turns stabilizer self-correction into an efficient learner for Clifford structure. In particular, given query access to a unitary U of Clifford fidelity at least τ, we can efficiently find a unitary V satisfying $|\langle\langle V|U\rangle\rangle|^2 \geq \mathrm{poly}(\tau)$. Together with boosting [AD26b, DJJ+26], this yields tomography algorithms for unitaries and Hamiltonians of bounded Clifford extent ξ, running in $\mathrm{poly}(n, \xi, 1/\varepsilon)$ time.[5]

In each case, the previous reductions were conditional on an efficient robust form of algorithmic PFR, or consequently inherited a quasipolynomial dependence on the relevant accuracy parameter. Our approximate algorithmic PFR theorem supplies the missing polynomial-time primitive and yields polynomial dependence on these parameters.

Acknowledgments. AG thanks Seyoon Ragavan for useful discussions, and the UK AISI alignment project for supporting access to ChatGPT Pro. AG is supported in part by a Simons investigator award and NSF grant CNS-2534400. SG is supported by the Herman Goldstine Memorial Postdoctoral Fellowship. SA and AD thank Jop Briet, Davi Castro Silva, and Tom Gur for several discussions surrounding this topic in the past.

AI Disclosure. The algorithm and proof strategy underlying the main theorem were suggested by ChatGPT 5.6 Sol. We found it crucial to prompt the model specifically to study the GGMT proof and search for a way to algorithmize its argument; more open-ended attempts to solve the problem without this direction were not successful. Subsequent interactions with AI were used to refine and clarify the proof. The authors independently verified, developed, and simplified the argument presented here and take full responsibility for its correctness, exposition, and attribution.

2 Technical overview

To motivate the algorithm underlying Theorem 1.2, we first revisit the proof of the PFR theorem due to Gowers, Green, Manners and Tao [GGMT24, GGMT25] (Section 2.1). Section 2.2 then discusses the main ideas necessary to algorithmize the existential proof.

[4] An n-qubit stabilizer state has stabilizer dimension k if it is stabilized by an Abelian group of 2^k Paulis.

[5] We write $|U\rangle\rangle \coloneqq (U \otimes I)\,|\mathrm{EPR}_n\rangle$, where $|\mathrm{EPR}_n\rangle$ consists of n EPR pairs.

2.1 The GGMT existential argument

An entropic formulation of the PFR theorem. The starting point of the proof by [GGMT25] is an entropic formulation of the PFR theorem, that deals with *random variables* instead of sets. Throughout, all random variables take values in $\mathbb{F}_2^n$. Given random variables X and Y, let X' and Y' be independent copies of X and Y, respectively. For a random variable X, its entropy is $\mathbf{H}[X] \coloneqq \sum_{x\in\mathbb{F}_2^n} \Pr[X=x] \log \frac{1}{\Pr[X=x]}$. The *entropic Ruzsa distance*[6] between X and Y is defined as

$$\mathrm{d}[X;Y] \coloneqq \mathbf{H}[X'+Y'] - \frac{1}{2}\mathbf{H}[X'] - \frac{1}{2}\mathbf{H}[Y'].$$

The entropic Ruzsa distance may be viewed as a measure of entropy growth under convolution. It is not a true metric—in particular, the self-distance $\mathrm{d}[X;X]$ measures the entropy growth produced by adding two independent samples from the same distribution, and need not be 0. It vanishes precisely when X is uniformly distributed on an affine subspace of G. Thus, small entropic self-distance indicates that X has approximate additive structure.

To connect the distributional notion with the usual combinatorial notion of small doubling, let $A \subseteq \mathbb{F}_2^n$ be nonempty and let U_A denote the uniform distribution on A. If $|A+A| \le K|A|$, then, for an independent copy U_A' of U_A,

$$\mathrm{d}[U_A;U_A] = \mathbf{H}[U_A+U_A'] - \mathbf{H}[U_A] \le \log|A+A| - \log|A| \le \log K,$$

using the trivial bound of $\mathbf{H}[D] \le \log \mathrm{supp}(D)$. Hence, a set with doubling constant at most K gives rise to a distribution with entropic self-distance at most $\log K$. The entropic formulation of PFR established by [GGMT25] states that for every $\mathbb{F}_2^n$-valued random variable X, there is a subspace $H \le \mathbb{F}_2^n$ such that

$$\mathrm{d}[X;U_H] = O\left(\mathrm{d}[X;X]\right),$$

where U_H denotes the uniform distribution on H. In this language, the PFR theorem is an *inverse theorem* for entropic Ruzsa distance: a distribution with small self-distance must be close, in entropic Ruzsa distance, to the uniform distribution on a subspace. See Theorem 3.5 for the formal statement.

An entropy-decrement argument. The proof of Gowers, Green, Manners, and Tao [GGMT25] is an iterative entropy-decrement argument. Starting from

$$X^{(0)} = Y^{(0)} = U_A, \qquad \mathrm{d}[X^{(0)};Y^{(0)}] \le \log K,$$

the aim is to repeatedly replace the current pair by one with smaller entropic Ruzsa distance, eventually reaching the structured regime. At the same time, one must control how far the distributions move along the way, otherwise the structured distribution found at the end could have little to do with the original distribution U_A. To balance these two objectives, given a current pair (X,Y), one measures the quality of a candidate update (X',Y') using the potential

$$\Phi_{X,Y}(X',Y') \coloneqq \mathrm{d}[X';Y'] + \eta\left(\mathrm{d}[X;X'] + \mathrm{d}[Y;Y']\right),$$

where $\eta > 0$ is a sufficiently small absolute constant. The first term measures how much additive complexity remains in the new pair, while the latter two terms penalize moving too far from the current pair. Thus, a good update must both reduce the entropic Ruzsa distance and remain quantitatively connected to the distributions from which it arose.

[6] Traditional Ruzsa calculus studies the *combinatorial* Ruzsa distance $\mathrm{d}_{\mathrm{comb}}[A;B] \coloneqq \log|A+B| - \frac{1}{2}\log|A| - \frac{1}{2}\log|B|$ between *sets* A, B. The entropic Ruzsa distance was first introduced by [Tao10]. It turns out that sets and the combinatorial Ruzsa distance are too rigid for the iterative argument in [GGMT25] to go through. A key proof idea of [GGMT25] is to instead work with *distributions* over sets and the entropic Ruzsa distance, to capture more flexible and fine-grained information.

The central decrement statement is as follows: whenever $\mathrm{d}[X;Y]$ is above a fixed absolute constant, GGMT showed that there are five explicit update families to the random variables (X, Y), one of which produces a pair (X', Y') satisfying

$$\mathbb{E}\left[\Phi_{X,Y}(X', Y')\right] \leq (1 - \Omega(1)) \cdot \mathrm{d}[X;Y].$$

Some of these updates involve randomness, and the expectation is over this randomness. All the five update families are built from two basic operations that we define shortly: *summing* and *fibering*. In a bit more detail, the five updates consist of self-sums, cross-sums, self-fibers, mixed fibers, and an "endgame" transformation consisting of first fibering and then summing.

Summing replaces the current variables (X, Y) by self-sums $(X+X', Y+Y')$ or cross-sums $(X+Y, X'+Y')$. The structured distributions we seek are *stable under addition*: if U_H is uniform on a subspace H, then $U_H + U_H \sim U_H$. For certain kinds of distributions resembling U_H, summing can achieve a large decrement in the entropic Ruzsa distance. For example, uniform distributions over dense random subsets $A \subseteq H$ that are contained within a subspace H, must remain within H, but expand to take up more room inside H, become closer to U_H and therefore decreasing the entropic Ruzsa distance.

However, there are other random variables for which summing *increases* the entropic Ruzsa distance. For example, consider (a uniform distribution over) the union of a small number of unevenly spaced cosets of a subspace $H \leq \mathbb{F}_2^n$. Summing is a bad idea here—it approximately doubles the entropic Rusza distance. Luckily, for such distributions, our second operation, *fibering*, effectively filters out a small number of cosets, making the distance drop to $O(1)$. Fibering can be thought of as "quotienting" operation that isolates algebraic structure in a way that complementary to summing: it restricts attention to preimages giving rise to a particular sum. Concretely, given a pair (X, Y), fibering outputs the conditional distribution $X|(X + Y) = z$, for a randomly sampled $z \sim X + Y$. We refer to z as the *fiber label*. The randomness in the decrement statement above comes from sampling these fiber labels according to their natural distribution.

Summing and fibering therefore provide two complementary ways to make progress. Summing studies global behavior of the current distributions under addition, whereas fibering searches locally for a structured component hidden inside an additive level set. Miraculously, [GGMT25] prove that these operations are sufficient. Four of five candidate updates arise directly from these operations: self-sums, cross-sums, self-fibers, and mixed fibers. Roughly speaking, either one of the relevant sums already exhibits lower additive complexity or one of its fibers reveals a more structured conditional distribution. The fifth, or *endgame*, update handles the case in which none of the four simpler transformations gives sufficient progress. Its analysis exploits additional structure forced by this failure; importantly for us, the resulting update can still be implemented using only the same primitives, namely two fiber operations followed by a sum.

So the high-level idea of GGMT is, starting from

$$X_0 = Y_0 = U_A, \qquad \mathrm{d}[X_0; Y_0] \leq \log K,$$

each favorable transformation decreases the entropic Ruzsa distance by a constant factor. After $O(\log \log K)$ favorable steps, the process reaches a pair at constant entropic Ruzsa distance, while the movement penalties ensure that the total distance traveled from U_A remains $O(\log K)$. For our purposes, the essential takeaway is that each step comes from five transformations built from sums and fibers, one of which makes substantial progress.

The next subsection explains how to implement this recursion efficiently without computing the relevant entropies or identifying which transformation is favorable.

2.2 Our algorithmic approach

The exposition above suggests a natural algorithmic strategy. Recall that the algorithm is given uniform sample access to an unknown set $A \subseteq \mathbb{F}_2^n$ with doubling constant at most K, together with a membership

oracle for A. Starting from $X_0 = Y_0 = U_A$, we would like to follow the GGMT recursion, repeatedly applying transformations that decrease the potential until the entropic Ruzsa distance becomes constant. Since each favorable transformation reduces the distance by a constant factor, only $O(\log\log K)$ successful transformations are needed. Turning this strategy into an efficient algorithm presents three main difficulties.

1. The GGMT proof identifies a favorable transformation through entropy inequalities, but estimating the relevant entropies of the intermediate distributions appears to be computationally infeasible.

2. The intermediate random variables are defined recursively through sums and fibers, and we need an efficient way to sample from and manipulate these distributions without ever writing down their probability mass function, which could have exponential support.

3. Once the recursion reaches constant entropic Ruzsa distance, we need to recover an explicit subspace from oracle access to the terminal distribution.

We address each of these difficulties in turn.

Random traversal of the GGMT tree. We begin with the first difficulty. For a current pair (X, Y), the decrement theorem of GGMT guarantees that one of the five GGMT update families has small average potential. Determining which family achieves the decrement, by estimating $\Phi_{X,Y}(X', Y')$, would require estimating entropies of distributions that may have exponentially large support. This appears computationally infeasible in general.[7]

Our algorithm therefore *never computes the entropy, entropic Ruzsa distance, or the potential.* Instead, at every step it chooses one of the five candidate update families uniformly at random, and samples any required fiber labels from their natural distributions. The decrement theorem of [GGMT25, GGMT24] shows that a random step has a constant positive probability of producing a constant-factor decrement. Consequently, a sequence of $O(\log\log K)$ favorable random steps reaches the constant-distance regime with probability $\exp\big(-O(\log\log K)\big) = 1/\mathrm{polylog}\, K$.

Implicit access to intermediate distributions. The second difficulty is maintaining algorithmic access to the random variables encountered along this traversal. Sampling access alone is not sufficient, since it does not allow us to continue the traversal. For example, suppose we wish to implement the fiber $X_z \coloneqq X \mid X + Y = z$. Writing p_X and p_Y for the probability mass functions of X and Y, we have

$$\Pr[X_z = x] = \Pr[X = x \mid X + Y = z] = \frac{p_X(x)p_Y(z-x)}{p_{X+Y}(z)}.$$

A natural way to sample from X_z is by rejection sampling: repeatedly sample $x \sim X$ and accept with probability proportional to $p_Y(z-x)$. A sampler for Y, however, gives no way to implement such an acceptance probability at a specified point.

We therefore augment sampling access with precisely the additional primitive needed for this rejection sampler as above: for every intermediate random variable X, we maintain not only an exact sampler Sample_X, but also a procedure Coin_X which, on input $x \in \mathbb{F}_2^n$, samples and outputs a bit satisfying

$$\Pr[\mathsf{Coin}_X(x) = 1] = \frac{p_X(x)}{M_X},$$

[7] At the root of the recursion, the self-distance of U_A can in fact be estimated efficiently in $\mathrm{poly}(n, K)$ time using uniform samples and membership queries. It is unclear how to extend this approach to the intermediate, generally nonuniform distributions, for which no analogous membership oracle is available.

where $M_X \geq \|p_X\|_\infty$ is an implicit normalization we call the *envelope*. Importantly, the envelope need not beknown to the algorithm. This fact turns out to be crucial in ensuring efficiency. Thus, Coin_X allows us to accept x with probability proportional to its probability mass $p_X(x)$, without ever explicitly evaluating $p_X(x)$. The pair $(\mathsf{Sample}_X, \mathsf{Coin}_X)$ provides our needed algorithmic access to X.

Importantly, this access is available at the root of the recursion directly from the input. For $X^{(0)}, Y^{(0)} = U_A$, the sampler Sample_{U_A} is given by the hypothesis, and the membership oracle serves as Coin_{U_A}, since we may take $M_{U_A} = 1/|A|$. Although $|A|$ is unknown to the algorithm, the corresponding coin satisfies

$$\Pr\big[\mathsf{Coin}_{U_A}(x) = 1\big] = \frac{p_{U_A}(x)}{M_{U_A}}.$$

Hence the membership oracle for A implements Coin_{U_A} exactly.

This form of access can be propagated exactly through both of the basic GGMT operations, summing and fibering. For a sum $Z = X + Y$, an exact sample is obtained by drawing $x \sim X$ and $y \sim Y$ independently and returning $x + y$. On input z, we may implement $\mathsf{Coin}_Z(z)$ by sampling $x \sim X$ and returning the outcome of $\mathsf{Coin}_Y(z - x)$, since

$$\Pr[\mathsf{Coin}_Z(z) = 1] = \sum_x p_X(x) \frac{p_Y(z-x)}{M_Y} = \frac{p_Z(z)}{M_Y}.$$

Thus, this is a valid coin for Z with implicit normalization $M_Z = M_Y$.

For a fiber $X_z = X \mid X + Y = z$, we repeatedly sample $x \sim X$ until $\mathsf{Coin}_Y(z - x)$ accepts, and return the first accepted sample. The probability that a particular x is returned is proportional to $p_X(x)p_Y(z-x)$, so the returned sample is distributed exactly according to X_z. A coin for X_z can be implemented by independently running $\mathsf{Coin}_X(x)$ and $\mathsf{Coin}_Y(z - x)$ and accepting if both succeed. Its acceptance probability is

$$\frac{p_X(x)p_Y(z-x)}{M_X M_Y} = \frac{p_{X_z}(x)}{M_X M_Y / p_{X+Y}(z)},$$

so it is again of the required form. Consequently, starting only from the sample and membership oracle for A, we can recursively construct exact algorithmic access to every intermediate random variable appearing in the GGMT recursion.

Bucketing to maintain efficient access. We have explained how to maintain the algorithmic access needed for every intermediate random variable appearing in the GGMT recursion. We have to ensure that this access can be implemented efficiently. The main difficulty comes from fibering. If $Z = X + Y$, then for a fiber label z, each sample $x \sim X$ is accepted by the rejection sampler with probability $p_Z(z)/M_Y$. Thus, the expected number of samples needed to obtain one sample from X_z is $M_Y/p_Z(z)$. To track how these rejection-sampling costs evolve through the recursion, we define the *slack* of our algorithmic access to X by

$$\Delta[X] \coloneqq \mathbf{H}[X] + \log M_X = \mathbb{E}_{x\sim X}\left[\log \frac{1}{r_X(x)}\right],$$

where we set $r_X(x) \coloneqq \Pr[\mathsf{Coin}_X(x) = 1]$ for convenience.[8] Thus, $\Delta[X]$ measures the typical logarithmic inverse acceptance probability of Coin_X on a sample $x \sim X$. Small slack means that Coin_X has reasonably large acceptance probability on a typical sample from X.

The efficiency of our representations can be quantified through the slack and the entropic Ruzsa distance. We start out with no slack,

$$\Delta[U_A] = \mathbf{H}[U_A] + \log \frac{1}{|A|} = 0.$$

[8]Note that the slack $\Delta[X]$ depends on the particular envelope M_X we use in the algorithmic representation of X, and, contrary to what the notation suggests, is not determined by X alone.

Although the root distribution has zero slack, sums and fibers can increase it. For sums, a direct calculation gives

$$\Delta[X+Y] \le \Delta[Y] + 2\mathrm{d}[X;Y],$$

and for a fiber $X_Z = X \mid X+Y=Z$,

$$\mathop{\mathbb{E}}_{Z}[\Delta[X_Z]] = \Delta[X] + \Delta[Y].$$

This is problematic, because for a fixed fiber label z, the rejection sampler requires $\frac{M_Y}{p_Z(z)}$ trials in expectation, and

$$\mathop{\mathbb{E}}_{z\sim Z}\left[\log \frac{M_Y}{p_Z(z)}\right] \le \Delta[Y] + 2\mathrm{d}[X;Y].$$

Thus, controlling the slack also controls the typical cost of sampling fibers.

To address this, we insert a *bucketing* step after each GGMT transformation.[9] Informally, bucketing conditions the current random variable on an approximate level set of the probability mass function, so that retained points have comparable values of $r_X(x)$. Crucially, the bucketing can itself be implemented using only the sampler and coin access described above. Given a random variable Z, we sample $z \sim Z$ and repeatedly run $\mathsf{Coin}_Z(z)$. If the first success occurs on trial T, we record $J := \lfloor \log_2 T \rfloor$ and condition on the value of J. Since T is concentrated around $1/r_Z(z)$, fixing $J=j$ effectively restricts Z to a probability scale on which $r_Z(z)$ is roughly 2^{-j}. The resulting conditional law Z_J again admits exact sample-and-coin access, and the key guarantee we show is $\mathbb{E}_J[\Delta[Z_J]] = O(1)$. Thus, bucketing regularly restores constant slack and prevents the rejection-sampling costs from accumulating throughout the recursion.

There is one additional concern: conditioning the intermediate distributions could interfere with the entropy decrement that drives the GGMT argument. Fortunately, conditioning a random variable Z by another random variable J can increase the entropic Ruzsa distance only by their mutual information, and bucketing reveals only

$$\mathbf{I}[Z;J] = O\big(\log \Delta[Z]\big)$$

bits of information.[10] At entropic Ruzsa distance d, the relevant slack is $O(d)$, so bucketing incurs only an $O(\log d)$ loss, whereas a good GGMT step yields a decrement of order $\Omega(d)$. Above a sufficiently large constant threshold, the decrement therefore dominates the cost of bucketing.

The total cost of a GGMT trajectory. Above a fixed constant threshold d_{term}, a random GGMT update followed by bucketing has constant probability of both achieving the desired potential decrement and producing a new pair with constant slack and controlled access cost. Once the distance falls below d_{term}, the same analysis gives a constant-probability update that keeps the distance, movement penalties, and slack bounded by absolute constants. We therefore run the recursion for a fixed $T = O(\log\log K)$ iterations without attempting to detect when the terminal regime is reached. With probability $1/\mathrm{polylog}\, K$, the resulting trajectory is good, and its terminal variable satisfies

$$\mathrm{d}[X_T;X_T] = O(1), \qquad \mathrm{d}[U_A;X_T] = O(\log(2K)), \qquad \Delta[X_T] = O(1).$$

It remains to bound the cost of accessing X_T. Let $d_t := \mathrm{d}[X_t, Y_t]$. On a good trajectory, the rejection-sampling bounds above imply that the sampler-and-coin cost grows from level t to level $t+1$ by at most a

[9] Our bucketing step is analogous to the probability bucketing used in distribution testing [BFF+01, CFMdW10], where the domain is partitioned into ranges of comparable probability mass so that the conditional distribution on each bucket is approximately flat.

[10] Strictly speaking, bounds such as $O(\log \Delta[Z])$ should be read as $O(\log(\Delta[Z]+2))$, so that the expression remains well behaved when $\Delta[Z]$ is small. We suppress such harmless additive constants inside logarithms here and throughout this paper.

factor $2^{O(d_t)}$. Hence, the cumulative overhead is

$$\prod_{t<T} 2^{O(d_t)} = 2^{O(\sum_{t<T} d_t)}.$$

Before reaching the terminal regime, the distances decrease by a constant factor from $d_0 \le \log K$; afterward, they remain $O(1)$. Since $T = O(\log\log K)$, we have $\sum_{t<T} d_t = O(\log K)$ an therefore the total overhead is

$$2^{O(\log K)} = K^{O(1)}.$$

Thus, along a good trajectory, both constructing the terminal access procedures and calling them have polynomial expected cost (Claim 4.14).

Extracting a candidate subspace. It remains to extract a PFR subspace from the terminal random variable X_T. A naive approach would be to sample from X_T and take the span, but even a few rare outliers can make this span much too large. Instead, we look at the self-sum $X_T + X_T'$. A value z that occurs with large probability in this self-sum has many likely representations $z = x + x'$, and therefore captures an additive direction that is genuinely present in the high-mass part of X_T. We use $\mathsf{Coin}_{X_T+X_T'}$ to identify these *heavy self-sums* and take the span of $O(n)$ independent heavy samples.

The constant self-distance and slack of X_T ensure that heavy self-sums carry constant mass, while the entropic PFR theorem implies that they are confined to a constant-index enlargement of an underlying subspace. Consequently, with constant probability their span V satisfies $\mathrm{d}[X_T; U_V] = O(1)$, and the triangle inequality gives $\mathrm{d}[U_A; U_V] = O(\log K)$. This subspace extraction phase uses only $O(n \log n)$ calls to the terminal access algorithms for X_T and polynomial additional computation.

Verifying a PFR subspace. A remaining issue is that the candidate subspace V can be larger than A. The distance bound implies $|V| \le K^{O(1)}|A|$. Thus, by shrinking V arbitrarily by $O(\log K)$ dimensions, we obtain a subspace $V' \le V$ such that either $V' = \{0\}$ or $|V'| \le |A|/2$. Importantly, this does not require knowing $|A|$. Moreover, for a uniform sample $a \sim U_A$, the coset $a + V'$ contains a $K^{-O(1)}$ fraction of A with constant probability.

To recognize successful candidates, we estimate

$$\alpha \coloneqq \frac{|A \cap (a + V')|}{|A|}, \qquad \beta \coloneqq \frac{|A \cap (a + V')|}{|V'|}.$$

We estimate α by sampling from A and testing membership in $a + V'$, and estimate β by sampling uniformly from $a + V'$ and testing membership in A. The first quantity measures the fraction of A captured by the coset, while the identity $\beta/\alpha = |A|/|V'|$ lets us certify that $|V'| \le |A|$ without knowing $|A|$. With appropriate margins, every accepted candidate has $|V'| \le |A|$ and captures a $K^{-O(1)}$ fraction of A, except with the allotted verification error probability.

Finally, these guarantees imply the desired covering property. Let $B \coloneqq A \cap (a + V')$, so that $|B| \ge K^{-O(1)}|A|$, and take a maximal family of pairwise disjoint translates $B + t_1, \dots, B + t_m$ with $t_i \in A$. Since each translate lies in $A + A$,

$$m|B| \le |A + A| \le K|A|,$$

and hence $m = K^{O(1)}$. By maximality, for every $t \in A$, the translate $B + t$ intersects some $B + t_i$. Thus there exist $b, b' \in B$ such that $b + t = b' + t_i$. Since b, b' lie in the same coset $a + V'$, we have $b' - b \in V'$, and therefore $t - t_i \in V'$. Hence

$$A \subseteq \bigcup_{i=1}^{m} (t_i + V'),$$

so A is covered by $K^{O(1)}$ translates of V'.

Amplifying success. A single trial consists of (1) traversing the GGMT tree to obtain a terminal random variable X_T, (2) extracting a candidate subspace V, and (3) verifying the resulting candidate. If the trial makes bad choices along the way, the rejection sampling can become too expensive, so we impose a sufficiently large polynomial cutoff time for traversal and extraction, after which the trial is aborted. A single trial succeeds with probability $\Omega(1/\mathrm{polylog}\, K)$. Repeating $\mathrm{polylog}(K)\log(2/\delta)$ independent trials, with verification error set appropriately, gives overall success probability at least $1-\delta$ and total running time $\mathrm{poly}(n, K, \log(1/\delta))$, proving Theorem 1.2.

3 Preliminaries

Notation and Terminology. Throughout this paper, we will work in the ambient space of $\mathbb{F}_2^n$. For $n \in \mathbb{N}$, we define $[n] := \{1, 2, \ldots, n\}$ and $[n]_0 := \{0\} \cup [n]$. All the random variables in this paper will take values in $\mathbb{F}_2^n$. For a non-empty finite set $A \subseteq G$, we let U_A denote the uniform distribution on A. All logarithms (denoted by log) are base 2 and we will specifically use ln to denote the natural logarithm. For a random variable X, we implicitly denote p_X to the distribution of the random variable X. Throughout this paper, we use the shorthand $\mathrm{polylog}(K)$ for $\log^{O(1)}(2K)$ to avoid weird behaviour when $K \in [1, 2]$, $O(\log K)$ as a shorthand for $O(\log K + 2)$ for similar reasons.

3.1 Additive combinatorics

For a set $A \subseteq \mathbb{F}_2^n$, we denote the sumset $A + A := \{a_1 + a_2 : a_1, a_2 \in A\}$, and more generally, we will use the sumset notation $\ell A := A + \ldots + A$ to denote the sum of ℓ copies of A.

Doubling constant. We call the ratio $|A+A|/|A|$ the *doubling constant* of A. The PFR theorem Theorem 1.1 tightly characterizes the doubling constant in terms of the algebraic structure it imposes, and is seen as a measure of how close A is to being a subgroup/subspace.

Additive energy. Another notion that captures approximate subgroup structure of a set is its *additive energy* $E(A)$, which has several equivalent expressions

$$\begin{aligned} E(A) &:= \{(a, b, c, d) : a, b, c, d \in A, a + b = c + d\} \\ &= \sum_{v \in \mathbb{F}_2^n} \left(\#\{(a, b) : a, b \in A, a + b = v\}\right)^2 \\ &= \sum_{v \in \mathbb{F}_2^n} |A \cap (A + v)|^2. \end{aligned}$$

It can then be shown that a set A with doubling constant K satisfies $E_4(A) \geq |A|^3/K$.

Ruzsa distance. For finite nonempty subsets A, B of $\mathbb{F}_2^n$, the (combinatorial) Ruzsa distance measures how much larger the sum/difference set $A + B$ is than the geometric mean of the sizes of A and B, so small Ruzsa distance means that A and B share strong additive structure. Formally, the Ruzsa distance is defined as

$$\mathrm{d}_{\mathrm{comb}}(A, B) = \log \frac{|A + B|}{\sqrt{|A||B|}}.$$

Although this is not a true distance, it satisfies non-negativity, symmetry and the triangle inequality.

3.2 Information theory

Entropy. The entropy $\mathbf{H}[X]$ of a G-valued random variable is defined as

$$\mathbf{H}[X] := \sum_{x \in G} p_X(x) \log \frac{1}{p_X(x)}, \tag{1}$$

where log is base 2, $p_X(x)$ is the probability distribution function of X.

The *conditional* entropy $\mathbf{H}[X|Y]$ of X relative to Y is given by

$$\mathbf{H}[X|Y] := \sum_{y} p_Y(y) \mathbf{H}[X|Y = y], \tag{2}$$

where y ranges over the support of p_Y. We quickly note that the chain rules of

$$\mathbf{H}[X, Y] = \mathbf{H}[X|Y] + \mathbf{H}[Y], \quad \mathbf{H}[X, Y|Z] = \mathbf{H}[X|Y, Z] + \mathbf{H}[Y|Z]. \tag{3}$$

Additionally, we have the following facts regarding entropy.

Fact 3.1. *Suppose Z is a nonnegative integer-valued random variable with mean μ. The geometric distribution then maximizes entropy of Z and*

$$\mathbf{H}[Z] \leq (\mu + 1) \log(\mu + 1) - \mu \log \mu.$$

Lemma 3.2. *For every discrete random variable Z, $\mathbf{H}[Z] \geq -\log\left(\sum_z p_Z(z)^2\right)$. Equivalently,*

$$\sum_z p_Z(z)^2 \geq 2^{-\mathbf{H}[Z]}.$$

Moreover, for every $\varepsilon > 0$, there is a set $T \subseteq \mathrm{supp}(Z)$ such that $\Pr[Z \in T] \geq 1 - \varepsilon$ and $|T| \leq 2^{\mathbf{H}[Z]/\varepsilon}$.

Proof. Since $-\log$ is convex, Jensen's inequality applied to the random variable $p_Z(Z)$ gives

$$\mathbf{H}[Z] = \mathbb{E}\left[\log \frac{1}{p_Z(Z)}\right] \geq -\log \mathbb{E}[p_Z(Z)] = -\log\left(\sum_z p_Z(z)^2\right).$$

Exponentiating proves the equivalent formulation. Finally, apply Markov's inequality to $\log(1/p_Z(Z))$ and take $T = \{z : \log(1/p_Z(z)) \leq \mathbf{H}[Z]/\varepsilon\}$. □

Mutual Information. The mutual information $\mathbf{I}[X : Y]$ is defined as

$$\mathbf{I}[X : Y] := \mathbf{H}[X] + \mathbf{H}[Y] - \mathbf{H}[X, Y]. \tag{4}$$

3.3 Entropic Ruzsa distance

The *entropic Ruzsa distance* between two independent G-valued random variables X, Y, denoted by $\mathrm{d}[X; Y]$, is defined as

$$\mathrm{d}[X; Y] := \mathbf{H}[X' + Y'] - \frac{1}{2}\mathbf{H}[X'] - \frac{1}{2}\mathbf{H}[Y'], \tag{5}$$

where X', Y' are independent copies of X, Y respectively. Despite it being called a distance, the Ruzsa distance is not a true notion of distance as $\mathrm{d}[X; X] = 0$ only when X is a coset of a subgroup [GGMT25]. It however satisfies the triangle inequality, as stated below.

Lemma 3.3 (Entropic Ruzsa triangle inequality)**.** *For G-valued random variables X, Y, Z,*

$$\mathrm{d}[X;Z] \le \mathrm{d}[X;Y] + \mathrm{d}[Y;Z]. \tag{6}$$

Lemma 3.4 (Basic entropy and Ruzsa-distance facts)**.** *Let X, Y be random variables on $\mathbb{F}_2^n$.*

1. *The entropy-difference bound is*
$$|\mathbf{H}[X] - \mathbf{H}[Y]| \le 2\mathrm{d}[X;Y].$$
2. *If φ is a homomorphism, then entropic Ruzsa distance contracts under φ:*
$$\mathrm{d}[\varphi(X);\varphi(Y)] \le \mathrm{d}[X;Y].$$
3. *If $V \le W \le \mathbb{F}_2^n$ are subspaces, then*
$$\mathrm{d}[U_V;U_W] = \frac{1}{2}\log\frac{|W|}{|V|}.$$

Proof. Since conditioning cannot increase entropy, $\mathbf{H}[X+Y] \ge \max\{\mathbf{H}[X], \mathbf{H}[Y]\}$, which proves the first assertion by expanding the definition of Ruzsa distance. The identity

$$2\mathrm{d}[X;Y] = \mathbf{I}[X : X+Y] + \mathbf{I}[Y : X+Y]$$

and data processing prove the second. For the third, observe that $U_V + U_W$ is uniform on W. □

Theorem 3.5 (Entropic PFR theorem [GGMT25])**.** *For every random variable X on $\mathbb{F}_2^n$, there is a subspace $H \le \mathbb{F}_2^n$ such that*

$$\mathrm{d}[X;U_H] = O(\mathrm{d}[X;X]).$$

Conditional entropic Ruzsa distance. We will also use the conditional variant of the entropic Ruzsa distance. If X, Y, Z, W are G-valued random variables, we define

$$\mathrm{d}[X \mid Z; Y \mid W] := \mathop{\mathbb{E}}_{z\sim Z, w\sim W} \mathrm{d}[(X \mid Z=z); (Y \mid W=w)], \tag{7}$$

If both X, Y are conditioned on the same variable, we write $\mathrm{d}[X; Y \mid Z] = \mathbb{E}_{z\sim Z}\,\mathrm{d}[X; (Y \mid Z=z)]$. We also have the following equivalent expression considering X', Y', Z', W' as independent copies of X, Y, Z, W respectively [GGMT25]

$$\mathrm{d}[X \mid Z; Y \mid W] := \mathbf{H}[X' + Y' \mid Z', W'] - \frac{1}{2}\mathbf{H}[X' \mid Z'] - \frac{1}{2}\mathbf{H}[Y' \mid W'] \tag{8}$$

Lemma 3.6. *Let (X, J) be jointly distributed, with X taking values in $\mathbb{F}_2^n$, and let Y be an $\mathbb{F}_2^n$-valued random variable independent of (X, J). Then*

$$\mathop{\mathbb{E}}_{j\sim J} \mathrm{d}[(X \mid J=j); Y] \le \mathrm{d}[X;Y] + \frac{1}{2}\mathbf{I}[X : J].$$

In particular, if J takes at most m values, then

$$\mathop{\mathbb{E}}_{j\sim J} \mathrm{d}[(X \mid J=j); Y] \le \mathrm{d}[X;Y] + \frac{1}{2}\log m.$$

Proof. Using an independent copy of Y in every conditional Ruzsa distance, the left-hand side equals

$$\begin{aligned}\mathbf{H}[X+Y \mid J] - \frac{1}{2}\mathbf{H}[X \mid J] - \frac{1}{2}\mathbf{H}[Y] &= \mathbf{H}[X+Y] - \mathbf{I}[X+Y : J] \\ &\quad - \frac{1}{2}\big(\mathbf{H}[X] - \mathbf{I}[X : J]\big) - \frac{1}{2}\mathbf{H}[Y] \\ &= \mathrm{d}[X;Y] - \mathbf{I}[X+Y : J] + \frac{1}{2}\mathbf{I}[X : J].\end{aligned}$$

Dropping the nonnegative mutual-information term proves the inequality. The final assertion follows from $\mathbf{I}[X : J] \le \mathbf{H}[J] \le \log m$. □

3.4 Boolean Fourier analysis and discrete derivatives

For $x, y \in \mathbb{F}_2^n$, let $\langle x, y\rangle := \sum_{i=1}^n x_i y_i \in \mathbb{F}_2$. All expectations below are uniform over the indicated vector space.

Fourier analysis. For $g : \mathbb{F}_2^n \to \mathbb{R}$, define its normalized Fourier transform by

$$\widehat{g}(\xi) := \mathop{\mathbb{E}}_y g(y)(-1)^{\langle \xi, y\rangle}, \qquad \xi \in \mathbb{F}_2^n.$$

We call g Boolean when $g : \mathbb{F}_2^n \to \{-1, 1\}$. Fourier inversion and Parseval's identity give

$$g(y) = \sum_{\xi \in \mathbb{F}_2^n} \widehat{g}(\xi)(-1)^{\langle \xi, y\rangle}, \qquad \mathop{\mathbb{E}}_y g(y)h(y) = \sum_{\xi \in \mathbb{F}_2^n} \widehat{g}(\xi)\widehat{h}(\xi).$$

In particular, $\sum_\xi \widehat{g}(\xi)^2 = \mathbb{E}_y\, g(y)^2$, which equals 1 when g is Boolean. We will also use the autocorrelation identity

$$\mathop{\mathbb{E}}_y g(y)g(y+x) = \sum_{\xi \in \mathbb{F}_2^n} \widehat{g}(\xi)^2(-1)^{\langle \xi, x\rangle}.$$

Consequently, if $H : \mathbb{F}_2^n \to \mathbb{R}$ has nonnegative Fourier coefficients, then Fourier inversion implies

$$H(w) \le H(0) \qquad (w \in \mathbb{F}_2^n).$$

Theorem 3.7 (Goldreich–Levin [GL89])**.** *Let $g : \mathbb{F}_2^n \to \{-1, 1\}$ and let $0 < \gamma, \delta \le 1$. There is a randomized algorithm which, given oracle access to g, outputs a list of $O(\gamma^{-2})$ frequencies such that, with probability at least $1 - \delta$, the list contains every $\xi \in \mathbb{F}_2^n$ satisfying $|\widehat{g}(\xi)| \ge \gamma$. The algorithm uses $\mathrm{poly}(n, 1/\gamma, \log(1/\delta))$ time and queries to g.*

Discrete derivatives. For $f : \mathbb{F}_2^n \to \mathbb{R}$ and $x \in \mathbb{F}_2^n$, the multiplicative discrete derivative of f in direction x is

$$f_x(y) := f(y)f(y+x).$$

The Gowers U^3 norm. The U^3-norm is defined by

$$\|f\|_{U^3}^8 := \mathop{\mathbb{E}}_{y, x_1, x_2, x_3} \prod_{\omega \in \{0,1\}^3} f(y + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3).$$

Repeated use of Parseval's identity gives

$$\|f\|_{U^3}^8 = \mathop{\mathbb{E}}_x \sum_{\xi \in \mathbb{F}_2^n} \widehat{f_x}(\xi)^4.$$

Helper lemmas. The following observation is adapted from Tulsiani–Wolf [TW14, Claim 4.15].

Claim 3.8 (Removing an affine offset). *For every $f : \mathbb{F}_2^n \to \mathbb{R}$, linear map $T : \mathbb{F}_2^n \to \mathbb{F}_2^n$, and $b \in \mathbb{F}_2^n$,*

$$\mathop{\mathbb{E}}_x \left|\widehat{f_x}(Tx+b)\right|^2 \le \mathop{\mathbb{E}}_x \left|\widehat{f_x}(Tx)\right|^2.$$

Proof. Define

$$F_T(z) := \mathop{\mathbb{E}}_x \left|\widehat{f_x}(Tx+z)\right|^2.$$

For $t \in \mathbb{F}_2^n$, the normalized Fourier coefficient of F_T at t is

$$\begin{aligned}
\widehat{F_T}(t) &= \mathop{\mathbb{E}}_z F_T(z)(-1)^{\langle t,z\rangle} \\
&= \mathop{\mathbb{E}}_{z,x,y,y'} f(y)f(y+x)f(y')f(y'+x)(-1)^{\langle Tx+z,y+y'\rangle+\langle t,z\rangle} \\
&= 2^{-n} \mathop{\mathbb{E}}_{x,y} f(y)f(y+x)f(y+t)f(y+t+x)(-1)^{\langle Tx,t\rangle} \\
&= 2^{-n} \mathop{\mathbb{E}}_{x,y} f_t(y)f_t(y+x)(-1)^{\langle T^{\mathsf{T}}t,x\rangle} \\
&= 2^{-n} \left|\widehat{f_t}(T^{\mathsf{T}}t)\right|^2 \ge 0.
\end{aligned}$$

Here the third equality follows because averaging over z forces $y + y' = t$, and the last equality follows by expanding the square of $\widehat{f_t}(T^{\mathsf{T}}t)$. Thus every Fourier coefficient of F_T is nonnegative. Fourier inversion therefore gives

$$F_T(b) = \sum_{t\in\mathbb{F}_2^n} \widehat{F_T}(t)(-1)^{\langle t,b\rangle} \le \sum_{t\in\mathbb{F}_2^n} \widehat{F_T}(t) = F_T(0),$$

which is the claimed inequality. □

Claim 3.9. *For a function $g : \mathbb{F}_2^n \to \{-1,1\}$, we have that*

$$\mathop{\mathbb{E}}_x \left|\mathop{\mathbb{E}}_y g(y)g(y+x)\right|^2 = \sum_{\xi\in\mathbb{F}_2^n} \widehat{g}(\xi)^4.$$

Proof. First, by Fourier inversion, we have that

$$\begin{aligned}
\mathop{\mathbb{E}}_y g(y)g(y+x) &= \mathop{\mathbb{E}}_y \sum_{\xi,\xi'\in\mathbb{F}_2^n} \widehat{g}(\xi)\widehat{g}(\xi')(-1)^{\langle y,\xi+\xi'\rangle+\langle x,\xi'\rangle} \\
&= \sum_{\xi,\xi'\in\mathbb{F}_2^n} \widehat{g}(\xi)\widehat{g}(\xi')(-1)^{\langle x,\xi'\rangle} \mathop{\mathbb{E}}_y (-1)^{\langle y,\xi+\xi'\rangle} \\
&= \sum_{\xi\in\mathbb{F}_2^n} \widehat{g}(\xi)^2(-1)^{\langle x,\xi\rangle},
\end{aligned}$$

where the final equality follows by the orthogonality of characters. Then, we have that

$$\begin{aligned}
\mathop{\mathbb{E}}_x \left|\mathop{\mathbb{E}}_y g(y)g(y+x)\right|^2 &= \mathop{\mathbb{E}}_x \left(\sum_{\xi\in\mathbb{F}_2^n} \widehat{g}(\xi)^2(-1)^{\langle x,\xi\rangle}\right)^2 \\
&= \mathop{\mathbb{E}}_x \sum_{\xi,\xi'\in\mathbb{F}_2^n} \widehat{g}(\xi)^2\widehat{g}(\xi')^2(-1)^{\langle x,\xi+\xi'\rangle} \\
&= \sum_{\xi\in\mathbb{F}_2^n} \widehat{g}(\xi)^4,
\end{aligned}$$

where again the final equality is by the orthogonality of characters. □

4 Traversing the GGMT tree

Throughout this section, we will need access to random variables for the algorithm and we formally define this in the definition below.

Definition 4.1 (Algorithmic access to a random variable)**.** *We say that we have* algorithmic access *to a random variable X if we have the following pair of algorithms*

- Sample_X *outputs an independent sample $x \sim X$ and runs in time expected time* $\mathrm{poly}(n, K)$*, and*
- Coin_X*, on input x, runs in expected time* $\mathrm{poly}(n, K)$ *and outputs an independent sample from the Bernoulli distribution with parameter*

$$r_X(x) := \frac{p_X(x)}{M_X},$$

where $M_X \geq \|p_X\|_\infty$ is referred to as the envelope.[11] *The numerical value of M_X need not be known to the algorithm, even approximately.*

The input oracles for A give us algorithmic access to the uniform distribution U_A on the set A. The given uniform sampler for A implements Sample_{U_A}, while the membership oracle implements the density coin

$$\mathsf{Coin}_{U_A}(x) = \mathbf{1}_A(x).$$

Indeed, taking $M_{U_A} := 1/|A|$ gives

$$\Pr\left[\mathsf{Coin}_{U_A}(x) = 1\right] = \mathbf{1}_A(x) = \frac{p_{U_A}(x)}{M_{U_A}}.$$

Neither root algorithm aborts, and the value of M_{U_A} need not be known to the algorithm. Throughout this section, by algorithmic access to X, we mean constructing algorithms for Sample_X and Coin_X. We define the ***slack*** of a density coin Coin_X, denoted by $\Delta[X]$, as

$$\Delta[X] := \mathbf{H}[X] + \log M_X. \tag{9}$$

For the root algorithmic access above, $\Delta[U_A] = 0$. Intuitively, the slack measures how small the density coin's acceptance probability is on a typical sample $x \sim X$. As we will see later in this section, a small slack means that the coin can typically be used efficiently, while large slack indicates that rejection-based procedures may require many trials. The main result of this section is the following.

Theorem 4.2. *Let $K \in \mathbb{N}$. Suppose $A \subseteq \mathbb{F}_2^n$ is non-empty set that satisfies $|A + A| \leq K|A|$. Given membership and uniform-sampling access to A, there is a randomized algorithm that, with probability at least $\Omega(1/\mathrm{polylog}(K))$, outputs algorithmic access to a random variable X_T, where $T = O(\log\log K)$, such that, for some subspace $H \leq \mathbb{F}_2^n$,*

$$\mathrm{d}[X_T; U_H] = O(1), \quad \mathrm{d}[U_A; X_T] = O(\log K), \quad \text{and } \Delta[X_T] = O(1).$$

The algorithm has overall sample and query complexity $\mathrm{poly}(K)$ and runs in time $O(n \cdot \mathrm{poly}(K))$. The cost of one call to the algorithmic access procedures of X_T has sample and query complexity $\mathrm{poly}(K)$ and time complexity $O(n \cdot \mathrm{poly}(K))$ as well.

We prove this theorem through the section. In Section 4.1, we explain the five operations involved in the GGMT tree. In Section 4.2, we give algorithmic implementations for these operations. In Section 4.3, we introduce bucketting to regularize the algorithmic access of the variables as we go through the GGMT tree and ensure that the slack of the variables is bounded. Finally in Section 4.4, we show how one can probabilistically sample a path in the GGMT tree to find the random variable X_T satisfying Theorem 4.2.

[11] Note that the envelope is determined by the particular implementation of the density coin, rather than being an intrinsic property of the law of X.

4.1 The five GGMT operations

Let X and Y be random variables over $\mathbb{F}_2^n$. Let X_0, X_1 be independent copies of X and let Y_0, Y_1 be independent copies of Y, with all four variables mutually independent. The five GGMT operations applied to (X, Y) are defined as follows: each operation should be thought of as taking a pair of distributions/random variables (X, Y) and producing a new pair of random variables (equivalently, their distribution laws).

1. *Self-sum:*
$$\mathsf{SelfSum}(X, Y) := (X_0 + X_1,\ Y_0 + Y_1).$$

2. *Cross-sum:*
$$\mathsf{CrossSum}(X, Y) := (X_0 + Y_0,\ X_1 + Y_1).$$

3. *Self-fiber:* First sample
$$x \sim (X_0 + X_1), \qquad y \sim (Y_0 + Y_1)$$
independently. We then restrict to the corresponding fibers of the addition map and define
$$\mathsf{SelfFiber}(X, Y) := ((X_0 \mid X_0 + X_1 = x), (Y_0 \mid Y_0 + Y_1 = y)).$$
Intuitively, the first argument is the distribution over a random variable defined as follows: among all pairs of X_0, X_1 whose sum equals x, how is the first argument distributed.

4. *Cross-fiber:* First sample
$$z_0 \sim (X_0 + Y_0), \qquad z_1 \sim (X_1 + Y_1)$$
independently. We then define
$$\mathsf{CrossFiber}(X, Y) := ((X_0 \mid X_0 + Y_0 = z_0), (Y_1 \mid Y_1 + X_1 = z_1)).$$
Here each fiber is obtained by conditioning on the sum of one copy of X and one copy of Y.

5. *Endgame:* Define
$$Z := X_1 + Y_0, \qquad W := X_0 + X_1 + Y_0 + Y_1.$$
Sample (z, w) according to the joint distribution of (Z, W), and let
$$U_{z,w} := (X_1 + Y_1 \mid Z = z,\ W = w).$$
We then set
$$\mathsf{Endgame}(X, Y) := (U_{z,w}, U_{z,w}).$$

We then have the following promise from [GGMT24] which we have written for the special case of $\mathbb{F}_2^n$.

Theorem 4.3. *There is a universal constant $\eta > 0$ such that at least one of these five raw candidate families, with (X', Y') drawn according to its natural law, satisfies*
$$\mathbb{E}[\mathsf{d}[X'; Y'] + \eta \cdot \mathsf{d}[X; X'] + \eta \cdot \mathsf{d}[Y; Y']] \le (1 - \eta)\mathsf{d}[X; Y].$$

For the self-sum and cross-sum families the output laws are deterministic, while for the other three families the expectation is over their fiber or endgame labels. We use this constant η for the remainder of the section.

4.2 Algorithmic access to candidate families

We now describe how the algorithmic access to (X_{t+1}, Y_{t+1}) is constructed given algorithmic access to (X_t, Y_t) and corresponding to each of the possible candidate decisions as described in Section 4.1.

4.2.1 Sums

We first give a procedure that given access to random variables X, Y allows to perform the sum operation (the first two components of the GGMT operations).

Lemma 4.4. *Suppose we have algorithmic access to X, Y. Then, there is a protocol that gives algorithmic access to $X + Y$ with envelope M_Y and slack satisfying*

$$\Delta[X + Y] \leq \Delta[Y] + 2\mathrm{d}[X; Y].$$

Proof. We construct algorithmic access to $X + Y$ as follows:

(*i*) Sample_{X+Y} runs $x \leftarrow \mathsf{Sample}_X, y \leftarrow \mathsf{Sample}_Y$ independently and then outputs $x + y$.

(*ii*) $\mathsf{Coin}_{X+Y}(z)$ samples $x \leftarrow \mathsf{Sample}_X$ and outputs $\mathsf{Coin}_Y(z - x)$.

The sampler is correct by definition. We now note that

$$p_{X+Y}(z) = \sum_x p_X(x)p_Y(z - x) \leq \sum_x p_X(x)M_Y = M_Y, \quad \forall z \in \mathbb{F}_2^n,$$

which implies that M_Y is an upper bound on $p_{X+Y}(z), \forall z$ and is a valid envelope for $X + Y$. We also note that Coin_{X+Y} gives us the desired output by evaluating

$$\Pr[\mathsf{Coin}_{X+Y}(z) = 1] = \frac{1}{M_Y} \sum_{x \in \mathbb{F}_2^n} p_X(x)p_Y(z - x) = \frac{p_{X+Y}(z)}{M_Y}.$$

where the equality follows from

$$p_{X+Y}(z) = \Pr[X + Y = z] = \sum_x \Pr[X = x,\ Y = z - x] = \sum_x p_X(x)p_Y(z - x),$$

which used the independence of X, Y. Finally, we compute the slack:

$$\Delta[X + Y] = \mathbf{H}[X + Y] + \log M_{X+Y} = \mathbf{H}[X + Y] + \log M_Y = \Delta[Y] + \mathbf{H}[X + Y] - \mathbf{H}[Y], \tag{10}$$

where we used the definition of $\Delta[Y]$ in the last equality. Since X and Y are independent,

$$\mathbf{H}[X + Y] \geq \mathbf{H}[X + Y \mid Y] = \mathbf{H}[X]. \tag{11}$$

From the definition of the entropic Ruzsa distance, we have

$$2\mathrm{d}[X; Y] = 2\mathbf{H}[X + Y] - \mathbf{H}[X] - \mathbf{H}[Y] \geq \mathbf{H}[X + Y] - \mathbf{H}[Y], \tag{12}$$

where we used Eq. (11) in the last inequality. Substituting the above into Eq. (10) gives us

$$\Delta[X + Y] \leq \Delta[Y] + 2\mathrm{d}[X; Y],$$

which proves the lemma statement. □

4.2.2 Fibers

We next give a procedure that given access to random variables X, Y allows to perform the fibering operation (the third and fourth components of the GGMT operations).

Lemma 4.5. *Suppose we have algorithmic access to independent X, Y. Let $Z := X + Y$. Then, for every fixed $z \in \mathrm{supp}(Z)$, there is a protocol that constructs algorithmic access to $X_z := (X \mid Z = z)$, with envelope $M_{X_z} := M_X M_Y / p_Z(z)$. A sample from Sample_{X_z} is accepted with probability $p_Z(z)/M_Y$, so the expected number of times we run the sampler is $M_Y/p_Z(z)$.*

Further, if the label is drawn as $z \sim X + Y$, then

$$\mathop{\mathbb{E}}_{z\sim Z}\left[\Delta(X_z)\right] = \Delta[X] + \Delta[Y], \tag{13}$$

$$\mathop{\mathbb{E}}_{z\sim Z}\left[\log\left(\frac{M_Y}{p_Z(z)}\right)\right] = \mathbf{H}[X+Y] + \log M_Y \le \Delta[Y] + 2\mathrm{d}[X;Y]. \tag{14}$$

Proof. For a fixed $z \in \mathrm{supp}(Z)$, we note that Baye's rule gives us

$$p_{X_z}(x) = \Pr[X = x | X + Y = z] = \frac{p_X(x)p_Y(z-x)}{p_Z(z)}. \tag{15}$$

We construct Sample_{X_z} using the following rejection sampler:

(i) Sample $x \leftarrow \mathsf{Sample}_X$.

(ii) Run $w \leftarrow \mathsf{Coin}_Y(z-x)$.

(iii) If $w = 1$, then return x **else** repeat from (i).

For Sample_{X_z}, the probability of outputting x (i.e., sampling and accepting x) on a single trial is $p_X(x)p_Y(z-x)/M_Y$. The total acceptance probability is then

$$\sum_x p_X(x)\Pr[\mathsf{Coin}_Y(z-x)=1] = \sum_x p_X(x)\frac{p_Y(z-x)}{M_Y} = \frac{p_Z(z)}{M_Y}.$$

Since we repeat trials in Sample_{X_z} until Coin_Y accepts, we have that conditioned on acceptance

$$\Pr\left[\mathsf{Sample}_{X_z} \text{ returns } x\right] = \frac{p_X(x)p_Y(z-x)/M_Y}{p_Z(z)/M_Y} = \frac{p_X(x)p_Y(z-x)}{p_Z(z)},$$

which matches the density of X_z from Eq. (15) as desired. This verifies the correctness of the above protocol for Sample_{X_z}.

We construct the density coin Coin_{X_z} on input x as follows:

(i) Run $w_1 \leftarrow \mathsf{Coin}_X(x)$ and $w_2 \leftarrow \mathsf{Coin}_Y(z-x)$.

(ii) Output $w_1 w_2$.

The coin $\mathsf{Coin}_{X_z}(x)$ thus only accepts exactly when both density coins in (i) succeed. In particular, Coin_{X_z} accepts with probability

$$\Pr\left[\mathsf{Coin}_{X_z}(x)=1\right] = \frac{p_X(x)}{M_X}\frac{p_Y(z-x)}{M_Y} = \frac{p_{X_z}(x)}{M_X M_Y/p_Z(z)},$$

where we used the expression of $p_{X_z}(x)$ from Eq. (15) in the seond equality. From the above expression, we then also have that the envelope is $M_{X_z} = M_X M_Y / p_Z(z)$.

We can show that Eq. (13) holds from direct evaluation:

$$\begin{aligned}\mathop{\mathbb{E}}_{z\sim Z} \Delta[X_z] &= \mathop{\mathbb{E}}_{z\sim Z} [\mathbf{H}[X_z] + \log M_X + \log M_Y - \log p_Z(z)] \\ &= \mathbf{H}[X \mid Z] + \log M_X + \log M_Y + \mathbf{H}[Z] \\ &= \mathbf{H}[X, Z] + \log M_X + \log M_Y \\ &= \mathbf{H}[X] + \mathbf{H}[Y] + \log M_X + \log M_Y,\end{aligned}$$

where we used the definition of $\mathbf{H}[X|Z], \mathbf{H}[Z]$ in the second line, definition of $\mathbf{H}[X, Z]$ in the third line and then noted that $Z = X + Y$ along with the fact that X, Y are independent random variables in the last line (i.e., $\mathbf{H}[X, Z] = \mathbf{H}[X, Y] = \mathbf{H}[X] + \mathbf{H}[Y]$). Finally, Eq. (14) follows

$$\mathop{\mathbb{E}}_{z\sim Z}[\log(M_Y / p_Z(z))] = \mathbf{H}[Z] + \log M_Y \le 2\mathrm{d}[X; Y] + \mathbf{H}[Y] + \log M_Y = 2\mathrm{d}[X; Y] + \Delta[Y],$$

where we used Eq. (12) in the second inequality after noting that $Z = X + Y$ by definition and the definition of $\Delta[Y]$ in the last inequality. This completes the proof. □

4.2.3 Endgame

Finally it remains to describe the algorithmic component of the final GGMT operation, i.e., the so-called endgame operation. We describe how to do that below.

Lemma 4.6. *Suppose we have algorithmic access to X, Y. Let X_0, X_1 be independent copies of X, let Y_0, Y_1 be independent copies of Y, and suppose all four variables are mutually independent. Define*

$$Z := X_1 + Y_0, \qquad S := X_0 + Y_1, \qquad W := Z + S.$$

For a fixed $(z, w) \in \mathrm{supp}(Z, W)$, let $s := z + w$ and

$$U_{z,w} := (X_1 + Y_1 \mid Z = z, W = w).$$

Then there is a protocol that constructs exact algorithmic access to $U_{z,w}$ with envelope $M_{U_{z,w}} = M_X M_Y / p_{X+Y}(s)$. The construction uses two fiber rejection samplers: one whose trial acceptance probability is $p_{X+Y}(z)/M_Y$, and one whose trial acceptance probability is $p_{X+Y}(s)/M_X$. Moreover, when (z, w) is drawn from the natural law of (Z, W),

$$\mathop{\mathbb{E}}_{(z,w)\sim(Z,W)} \Delta[U_{z,w}] \le \Delta[X] + \Delta[Y] + 4\mathrm{d}[X; Y], \tag{16}$$

$$\mathop{\mathbb{E}}_{(z,w)\sim(Z,W)} \left[\log \frac{M_Y}{p_{X+Y}(z)} + \log \frac{M_X}{p_{X+Y}(w+z)}\right] = \Delta[X] + \Delta[Y] + 2\mathrm{d}[X; Y]. \tag{17}$$

Proof. Consider a fixed $(z, w) \in \mathrm{supp}(Z, W)$ and set $s := w + z$. We then define

$$F_z := (X_1 \mid X_1 + Y_0 = z), \qquad G_s := (Y_1 \mid X_0 + Y_1 = s).$$

Since the pairs (X_1, Y_0) and (X_0, Y_1) are independent, Z and S are independent. Moreover, the condition $Z = z$ and $W = w$ (or $Z + S = z + s$) is then equivalent to $Z = z$ and $S = s$. Consequently, $U_{z,w} := (X_1 + Y_1 | Z = z, W = w)$ can be equivalently written as

$$U_{z,w} = F_z + G_s,$$

where the two summands on the right are independent. To implement algorithmic access to $U_{z,w}$, we first need algorithmic access to F_z and G_s. We implement algorithmic access to F_z using Lemma 4.5 and denote the corresponding algorithms as Sample_{F_z} and Coin_{F_z}. We note that a single trial from Sample_{F_z} is accepted with probability $p_{X+Y}(z)/M_Y$ and the envelope is $M_{F_z} = M_X M_Y / p_{X+Y}(z)$. Similarly, we implement algorithmic access to G_s again using Lemma 4.5 and denote the corresponding algorithms as Sample_{G_s} and Coin_{G_s}. A single trial from Sample_{G_s} is accepted with probability $p_{X+Y}(s)/M_X$ and the envelope is $M_{G_s} = M_X M_Y / p_{X+Y}(s)$. Algorithmic access to $U_{z,w}$ is then implemented using Lemma 4.4 with F_z and G_s. The corresponding envelope is simply

$$M_{U_{z,w}} = M_{G_s} = M_X M_Y / p_{X+Y}(s).$$

It remains to prove the averaged bounds on the slack of $U_{z.w}$ and the acceptance probabilities involved as part of the construction of algorithmic access to $U_{z,w}$. Upon direct evaluation:

$$\begin{aligned}
\mathop{\mathbb{E}}_{(z,w)\sim(Z,W)} \Delta[U_{z,w}] = \mathop{\mathbb{E}}_{(z,s)\sim(Z,S)} \Delta[F_z + G_s] &\leq \mathop{\mathbb{E}}_{(z,s)\sim(Z,S)} \Delta[G_s] + 2 \mathop{\mathbb{E}}_{(z,s)\sim(Z,S)} \mathrm{d}[F_z; G_s] \\
&\leq \mathop{\mathbb{E}}_{s\sim S} \Delta[G_s] + 2 \mathop{\mathbb{E}}_{(z,s)\sim(Z,S)} \mathrm{d}[F_z; G_s] \\
&\leq \Delta[X] + \Delta[Y] + 2 \mathop{\mathbb{E}}_{(z,s)\sim(Z,S)} \mathrm{d}[F_z; G_s], \qquad (18)
\end{aligned}$$

where we have used the earlier observation $Z = z, W = w$ is equivalent to $Z = z, S = s$ in the first equality in the first line, and used Lemma 4.4 in the second equality in the first line to comment $\Delta[F_z + G_s] \leq \Delta[G_s] + 2\mathrm{d}[F_z; G_s]$. In the second line, we used the fact that Z and S are independent and both have the law of $X + Y$. In the final line, we used Lemma 4.5 from which we have

$$\mathop{\mathbb{E}}_{s\sim S} \left[\Delta[G_s]\right] = \Delta[X] + \Delta[Y]$$

We now bound the last term on the right in Eq. (18). Let us define $h := \mathbf{H}[X] + \mathbf{H}[Y] - \mathbf{H}[X + Y]$ and note that

$$\mathbf{H}[X_1 \mid Z] = \mathbf{H}[X_1, Z] - \mathbf{H}[Z] = \mathbf{H}[X_1, Y_0] - \mathbf{H}[Z] = \mathbf{H}[X] + \mathbf{H}[Y] - \mathbf{H}[X + Y] = h, \qquad (19)$$

where we have used Eq. (3) in the first equality, noted that the $Y_0 = Z + X_1$ and hence they determine each other in the second equality, and used the fact that X_1, Y_0 are independent along with the definition of Z in the final equality. Similarly, we can show that

$$\mathbf{H}[Y_1 \mid S] = \mathbf{H}[Y_1, S] - \mathbf{H}[S] = \mathbf{H}[Y_1, X_0] - \mathbf{H}[S] = \mathbf{H}[X] + \mathbf{H}[Y] - \mathbf{H}[X + Y] = h. \qquad (20)$$

Using the definition of conditional entropic Ruzsa distance (Eq. (7)) and its equivalent expression (Eq. (8)), we have that

$$\mathop{\mathbb{E}}_{z,s} \mathrm{d}[F_z; G_s] = \mathbf{H}[X_1 + Y_1 \mid Z, S] - \frac{1}{2}\mathbf{H}[X_1 \mid Z] - \frac{1}{2}\mathbf{H}[Y_1 \mid S] \leq \mathbf{H}[X + Y] - h = 2\mathrm{d}[X; Y] \qquad (21)$$

where we have used the fact that conditioning cannot increase entropy along with Eqs (19)–(20) in the first equality, and the definition of $\mathrm{d}[X; Y]$ in the last equality. Substituting Eq. (21) into Eq. (18) gives us the desired result of Eq. (16).

Finally, to obtain Eq. (17), we observe upon direct evaluation that

$$\mathop{\mathbb{E}}_{(z,s)\sim(Z,S)} \left[\log \frac{M_Y}{p_{X+Y}(z)} + \log \frac{M_X}{p_{X+Y}(s)}\right] = \mathop{\mathbb{E}}_{z\sim Z} \log \frac{M_Y}{p_{X+Y}(z)} + \mathop{\mathbb{E}}_{s\sim S} \log \frac{M_X}{p_{X+Y}(s)} \leq \Delta[X] + \Delta[Y] + 2\mathrm{d}[X; Y],$$

where the first equality follows from Z, S being independent and the last inequality follows from application of Eq. (14) of Lemma 4.5. This completes the proof. □

4.3 Bucketing to regularize access

So far, we showed that algorithmic access to all the intermediate random variables used in GGMT operations is possible given access to random variables in the iterative process. However, we also need to control the quantitative *quality* of this algorithmic representation as the recursion proceeds. To this end, recall that the *slack* of our algorithmic access to a random variable X is

$$\Delta[X] := \mathbf{H}[X] + \log M_X = \mathop{\mathbb{E}}_{x\sim X}\left[\log \frac{1}{r_X(x)}\right], \qquad r_X(x) := \Pr[\mathsf{Coin}_X(x) = 1] = \frac{p_X(x)}{M_X}.$$

Intuitively, slack measures the typical inverse acceptance probability of the density coin, i.e., measures how small the density-coin acceptance probability typically is on a sample $x \sim X$. So $\Delta[X] = O(1)$ means that, for a typical sample from X, the coin acceptance probability $r_X(x)$ is not exponentially small. This is exactly the quantity relevant to the rejection-sampling procedures used later in the recursion. Since fiber sampling is implemented by rejection sampling using that coin, large slack means the next fiber can require exponentially many trials cost of the rejection samplers used in subsequent fiber operations.

At the root of the recursion, $X = U_A$ and $\Delta[U_A] = 0$, so the density-coin representation starts with zero slack. The main issue in the GGMT operations is fibering: sampling a fiber is implemented by rejection sampling, whose cost is governed by the acceptance probability of the current density coin. While this cost is controlled for a single typical fiber, successive fiber operations can cause the slack of our representation to accumulate, making later rejection samplers increasingly expensive. Equivalently, the density coins of the intermediate distributions may have smaller and smaller acceptance probabilities on typical samples, making subsequent rejection samplers increasingly expensive. Since a GGMT trajectory contains several such operations, controlling the cost of each operation in isolation is not enough.

To prevent this accumulation, after every GGMT transformation we apply a *bucketing* operation. Informally, bucketing conditions the current random variable on a coarse dyadic estimate of the acceptance probability $r_X(x)$. Within a selected bucket, the values of $r_X(x)$ are all on approximately the same scale, and this regularizes the density coin and restores constant slack on average. The key point is that this computational regularization is cheap from the information-theoretic point of view: the bucket label reveals only $O(\log \Delta[X])$ bits of information and furthermore the new slack is only a constant. Thus bucketing can restore efficient algorithmic access keeping slack constant and without losing more than a logarithmic amount in the entropic Ruzsa-distance potential (which we show in the next section).

Lemma 4.7. *Given algorithmic access to X, there is a random variable J taking nonnegative integer values such that $\mathbf{I}[X : J] = O(\log \Delta[X])$. Moreover, there is an algorithm that constructs algorithmic access to $X_j := (X \mid J = j)$ with envelope and expected slack satisfying*

$$\mathop{\mathbb{E}}_{j\sim J}[\Delta[X_j]] = O(1), \text{ and } M_{X_j} = M_X/(2^j \cdot \Pr[J = j]),$$

respectively. Further, Sample_{X_j} uses an expected number of parent calls $O(2^{j+1}/\Pr[J = j])$,[12] and one call to Coin_{X_j} uses at most 2^{j+1} parent coins.

Bucketing algorithm. Before giving the proof of the above lemma, let us introduce some relevant notation and the underlying algorithm. We recall that Coin_X accepts x with probability $r_X(x) = p_X(x)/M_X$ and the slack is $\Delta[X] = \mathbb{E}_{x\sim x} \log(1/r_X(x))$. We would like to regularize X to have constant slack by conditioning X on a coarse estimate of $\log(1/r_X(x))$. As $r_X(x)$ for a fixed $x \in \mathrm{supp}(X)$ is not known from the algorithmic access to X, we need to obtain an estimate of this quantity. This can be done by sampling

[12] Here, "parent calls" refers to calls to the algorithmic access procedures for the original random variable X before bucketing, namely Sample_X and Coin_X.

$x \sim X$, then repeatedly running $\mathsf{Coin}_X(x)$ and letting N_X be the first successful trial. Conditioned on $X = x$, we note that N_x is geometric with mean $1/r_X(x)$ i.e., $N_X \sim \mathsf{Geom}(r_X(x))$ and $\mathbb{E}[N_X | X = x] = 1/r_X(x)$.

Let us define the random variable J which takes value $J = j$ when $2^j \le N_X < 2^{j+1}$ or in other words $J := \lfloor \log N_X \rfloor$ is a dyadic estimate of $\log(1/r_X(x))$. We then define

$$X_j := (X \mid J = j) = (X \mid 2^j \le N_X < 2^{j+1}), \quad \text{and} \quad \pi_j := \Pr[J = j]. \tag{22}$$

The goal is to then implement algorithmic access to X_j, which is the desired regularized version of X. The above protocol for obtaining N_X and implementing access to X_j is described in Algorithm 1, which we will use for Lemma 4.7.

Algorithm 1: Bucket(X)

Input: Algorithmic access to X (Definition 4.1)

Output: Label j, algorithmic access to X_j

1 Sample $x \leftarrow \mathsf{Sample}_X$.
2 Run $\mathsf{Coin}_X(x)$ until the first success and denote N_x as that time.
3 Set $j := \lfloor \log N_x \rfloor$.
4 Define **function** Sample_{X_j}
5 Sample $x \leftarrow \mathsf{Sample}_X$ and draw independent $\mathsf{Coin}_X(x)$ outcomes
6 If the first $2^j - 1$ coins fail and at least one of the next 2^j coins succeeds, output x.
7 Otherwise repeat from step 5.
8 Define **function** Coin_{X_j}
9 Draw 2^{j+1} independent $\mathsf{Coin}_X(x)$ outcomes.
10 Accept if the first 2^j trials contain exactly one success and the next 2^j trials contain at least one success.
11 **Return** $(j, \mathsf{Sample}_{X_j}, \mathsf{Coin}_{X_j})$.

Guarantees of bucketing. We are now ready to provide a proof of Lemma 4.7 by first showing the correctness of the implementation of algorithmic access to X_j in Algorithm 1 and then proving two information-theoretic bounds.

Claim 4.8. *Consider the context of Lemma 4.7. Algorithm 1 implements algorithmic access to X_j, with envelope $M_{X_j} = M_X/(2^j \Pr[J = j])$, for j sampled as in line 3. Sample_{X_j} uses an expected number of $O(2^{j+1}/\Pr[J = j])$ calls to the algorithmic access of X and Coin_{X_j} uses at most 2^{j+1} calls to Coin_X.*

Proof. Let us denote $r := r_X(x)$ and $m = 2^j$. Conditioned on $X = x$, the probability of the event $J = j$ (or $2^j \le N_x < 2^{j+1}$) is then

$$\Pr[J = j | X = x] := (1 - r)^{m-1}\big(1 - (1 - r)^m\big). \tag{23}$$

Using the notation of $\pi_j := \Pr[J = j]$, Bayes' rule then gives us that

$$p_{X_j}(x) = \frac{p_X(x) \Pr[J = j | X = x]}{\pi_j}. \tag{24}$$

We implement Sample_{X_j} as given in step 4 of Algorithm 1. We note that one trial of the rejection-sampling loop accepts with precisely the following probability as check if the first $m - 1$ parent coins fail after

sampling $x \sim X$ and require at least one success among the next m parent coins

$$\sum_x p_X(x) \Pr[J = j | X = x] = \pi_j$$

Conditioned on acceptance, it then returns x with probability p_{X_j} as given in Eq. (24). The sampler is thus exact and the expected number of trials is $1/\pi_j$. Each iteration uses fewer than 2^{j+1} parent coins. This proves the claimed $O(2^{j+1}/\pi_j)$ expected parent-call bound.

We implement Coin_{X_j} using step 8 of Algorithm 1. The goal is to ensure that $\Pr\left[\mathsf{Coin}_{X_j}(x) = 1\right]$ is proportional to $p_{X_j}(x)$ in Eq. (24) with an appropriately defined envelope. We note that the probability of acceptance associated with line 10 is

$$\Pr\left[\mathsf{Coin}_{X_j}(x) = 1\right] = mr(1-r)^{m-1}\left(1 - (1-r)^m\right) = mr \Pr[J = j | X = x] = \frac{m\pi_j}{M_X} p_{X_j}(x),$$

where we have used Eq. (23) in the second equality, and used the expression of $p_{X_j}(x)$ (Eq. (24)) along with the definition of $r = r_X(x) = p_X(x)/M_X$ in the final equality. Noting that the above is a valid probability law and is true $\forall x$, we then have that X_j has an envelope $M_{X_j} = M_X/(2^j\pi_j)$. This verifies the correctness of Coin_{X_j}. The cost of one call to Coin_{X_j} is the 2^{j+1} calls to Coin_X to carry out step 8. This completes the proof. □

We now show that conditioning X on J leads to suppression of the slack and regularizes the access.

Claim 4.9. *Consider the context of Lemma 4.7. Algorithm 1 implements algorithmic access to X_j with slack satisfying $\mathbb{E}_{j\sim J}[\Delta[X_j]] < 8$ and $\mathbf{I}[X : J] \le \log(3\Delta[X] + 3)$.*

Proof. We note that the expected slack of X_j is given by

$$\begin{aligned}
\mathop{\mathbb{E}}_{j\sim J}[\Delta[X_j]] &= \mathop{\mathbb{E}}_{j\sim J}\left[\mathbf{H}[X_j] + \log M_X - j - \log \Pr[J = j]\right] \\
&= \mathbf{H}[X \mid J] + \log M_X - \mathbb{E}[J] + \mathbf{H}[J] \\
&= \mathbf{H}[J \mid X] + \log M_X - \mathbb{E}[J] + \mathbf{H}[X] \\
&= \mathbf{H}[J \mid X] + \log M_X - \mathop{\mathbb{E}}_X\Big[\mathbb{E}[J \mid X]\Big] + \mathop{\mathbb{E}}_X[\log(1/p_X)] \\
&= \mathbf{H}[J \mid X] - \mathop{\mathbb{E}}_X\Big[\mathbb{E}[J \mid X]\Big] + \mathop{\mathbb{E}}_X[\log(1/r_X)], \qquad (25)
\end{aligned}$$

where we have used the definition of slack (Eq. (9)) and Claim 4.8 in the first line, definition of conditional entropy in the second line, the entropy chain rule of Eq. (3) to say $\mathbf{H}[X \mid J] + \mathbf{H}[J] = \mathbf{H}[X] + \mathbf{H}[J \mid X]$ in the third line, and definition of r_X in the last line. We now bound the three terms in Eq. (25) separately.

Let us consider a fixed $x \in \mathrm{supp}(X)$. We define $r := r_X(x)$ and $k := \lfloor \log(1/r) \rfloor$. Since N_x is geometric with parameter r, we have that

$$\Pr[J \ge j \mid X = x] = \Pr\left[N_x \ge 2^j\right] = (1-r)^{2^j - 1}. \qquad (26)$$

For $j \le k$, we also have

$$\Pr[J < j \mid X = x] \le \sum_{i=1}^{2^j-1} \Pr[N_X = i \mid X = x] = \sum_{i=1}^{2^j-1} (1-r)^{i-1} r \le r2^j, \qquad (27)$$

where we applied the union bound in the first inequality. We now bound $\mathbb{E}[J \mid X = x]$. Noting that for every non-negative integer-valued random variable Z, we have $\mathbb{E}\, Z = \sum_{j \geq 1} \Pr[Z \geq j]$, we have that

$$\begin{aligned}\mathbb{E}[J \mid X = x] = \sum_{j=1}^{\infty} \Pr[J \geq j \mid X = x] &\geq \sum_{j=1}^{k} \Pr[J \geq j \mid X = x] \\ &\geq \sum_{j=1}^{k} (1 - \Pr[J < j \mid X = x]) \\ &\geq \sum_{j=1}^{k} (1 - r2^{j}) \\ &= k - r(2^{k+1} - 2) > k - 2 \geq \log(1/r) - 3, \end{aligned} \tag{28}$$

where we have used Eq. (27) in the third line and definition of $k = \lfloor \log(1/r) \rfloor$ in the last inequality. We can upper bound $\mathbb{E}[J \mid X = x]$ as

$$\mathbb{E}[J \mid X = x] = \mathbb{E}[\log N_X \mid X = x] \leq \log \mathbb{E}[N_X \mid X = x] = \log(1/r), \tag{29}$$

where we have used Jensen's inequality in the second inequality and the fact that N_x has mean $\log(1/r)$ in the last equality.

We now bound the conditional entropy $\mathbf{H}[J \mid X]$. Since $k = \lfloor \log(1/r) \rfloor$, we equivalently have $2^{-(k+1)} < r \leq 2^{-k}$. For an integer $s \in [1, k]$, we have that

$$\Pr[J \leq k - s \mid X = x] = \Pr[J < k - s + 1 \mid X = x] \leq r2^{k-s+1} \leq 2^{1-s}, \tag{30}$$

where we applied Eq. (27) in the second inequality and used $r \leq 2^{-k}$ in the last inequality. Note for $s > k$, we trivially have that the upper bound is 0 since $J \geq 0$. To obtain an upper tail bound, consider any integer $s \geq 0$ and note that

$$\Pr[J \geq k + s \mid X = x] = (1 - r)^{2^{k+s}-1} \leq \exp\Big(-r(2^{k+s} - 1)\Big) \leq \exp\big(1 - 2^{s-1}\big), \tag{31}$$

where we used $1 - u \leq \exp(-u), \forall u \geq 0$. By a union bound and using Eqs. (30)–(31), we then have

$$\Pr[|J - k| \geq s \mid X = x] \leq 2^{1-s} + \exp\big(1 - 2^{s-1}\big). \tag{32}$$

Using the fact that for every non-negative integer-valued random variable Z, we have $\mathbb{E}\, Z = \sum_{j \geq 1} \Pr[Z \geq j]$, we can then evaluate

$$\mathbb{E}[|J - k| \mid X = x] = \sum_{s=1}^{\infty} \Pr[|J - k| \geq s \mid X = x] \leq \sum_{s=1}^{\infty} 2^{1-s} + \sum_{s=1}^{\infty} \exp\big(1 - 2^{s-1}\big) < 2 + 3/2 < 4, \tag{33}$$

where we used Eq. (32) in the second inequality and the fact that $2^{s-1} \geq 2(s-1), \forall s \geq 2$ in the penultimate inequality. We now obtain

$$\begin{aligned}\mathbf{H}[J \mid X = x] = \mathbf{H}[J - k \mid X = x] &\leq 1 + \mathbf{H}[|J - k| \mid X = x] \\ &< 1 + (1 + \mathbb{E}[|J - k| \mid X = x) \log(1 + \mathbb{E}[|J - k| \mid X = x) < 5, \end{aligned} \tag{34}$$

where the entropy remains unchanged in the first equality since k is deterministic, and the second inequality follows from noting that $J - k$ is determined by $|J - k|$ and one sign bit. The third equality follows from

Fact 3.1 and the final inequality then follows from Eq. (33). We can now bound the expected slack from Eq. (25)

$$\mathop{\mathbb{E}}_{j\sim J}[\Delta[X_j]] = \mathbf{H}[J \mid X] - \mathop{\mathbb{E}}_{X}\Big[\mathbb{E}[J \mid X]\Big] + \mathop{\mathbb{E}}_{X}[\log(1/r_X)] \le 5 + 3 - \mathop{\mathbb{E}}_{X}[\log(1/r_X)] + \mathop{\mathbb{E}}_{X}[\log(1/r_X)] = 8, \quad (35)$$

where we used Eq. (34) to bound the first term and Eq. (28) to bound the second term in the second inequality. This gives us the desired result regarding the slack.

To bound $\mathbf{I}[X : J]$, we note that

$$\mathbb{E}[J] = \mathop{\mathbb{E}}_{X}\Big[\mathbb{E}[J \mid X]\Big] \le \mathop{\mathbb{E}}_{x\sim X} \log \frac{1}{r_X(x)} = \mathbf{H}[X] + \log M_X = \Delta[X], \quad (36)$$

where the second inequality follows from Eq. (29) and then evaluate

$$\mathbf{I}[X : J] \le \mathbf{H}[J] \le (\mathbb{E}[J] + 1)\log(\mathbb{E}[J] + 1) - \mathbb{E}[J]\log\mathbb{E}[J] \le \log(e\,\mathbb{E}[J] + e) = \log(3\Delta[X] + 3), \quad (37)$$

where we used Fact 3.1. This completes the proof. □

Combining Claim 4.8 and Claim 4.9 gives us the proof of the desired Lemma 4.7.

4.4 Sampling a trajectory from the GGMT tree

We have so far shown how to implement algorithmic access for each of the possible candidate families (Section 4.1) at every iteration of the GGMT tree exactly. Moreover, when there was an approximate post-selection involved (such as in the fibering steps), we showed that bucketing prevents the slack (defined in Eq. (9)) of intermediate random variables from accumulating (Lemma 4.7) and hence can still be implemented efficiently. What remains is to navigate the GGMT tree itself. In the original combinatorial proof, one uses entropic inequalities to identify a favorable update and continues until the entropic Ruzsa distance becomes small.

We now explain how to navigate the GGMT recursion (Theorem 4.3) without ever computing an entropy or identifying which candidate gives the desired decrement (which is what the combinatorial argument of GGMT did). At this point the observation we have is simple, the depth of the GGMT tree is $O(\log\log K)$ and the tree at each layer has fanout size 5 [13], so one can potentially just *enumerate* all possible leaves of this tree with just a $\mathrm{polylog}(K)$ blow up. Our idea then is to simply choose among the five candidate families at random in each iteration. When the current entropic Ruzsa distance $d_t := \mathrm{d}[X_t; Y_t]$ is above some specified terminal threshold d_{term}, we will show that the GGMT decrement theorem (Theorem 4.3) together with bucketing (Lemma 4.7) implies that an iteration has constant probability of simultaneously producing a constant-factor decrease in d_t, restoring constant slack, and incurring cost at most $2^{O(d)} = \mathrm{polylog}(K)$.

Concretely, this leads to the following algorithm (Algorithm 2) that we will use to prove Theorem 4.2.

[13] This glosses over the fact that some of the candidate operations involve fiber sampling which have 2^n support. However, we will show a large fraction of these labels are favorable.

Algorithm 2: Sampling trajectories from the GGMT tree

Input: Sample access to U_A, query access to 1_A, doubling constant K

Output: Algorithmic access to X_T (Definition 4.1) satisfying Theorem 4.2 w.p. $\Omega(1/\mathrm{polylog}\, K)$.

Initialize $X_0 = Y_0 = U_A$.

Set maximum number of iterations $T = O(\log\log K)$ (Claim 4.13).

Set complexity budgets $B_S = \mathrm{poly}(K)$, $B_Q = \mathrm{poly}(K)$, and $B_{TC} = O(n \cdot \mathrm{poly}(K))$ (Claim 4.15).

Initialize budget counters $N_S = N_Q = N_{TC} = 0$.

for $t = 1$ **to** T **do**

  Obtain X'_t, Y'_t from X_{t-1}, Y_{t-1} by choosing one of the five GGMT candidates (Section 4.1) uniformly at random.

  Construct algorithmic access to X'_t, Y'_t using the corresponding lemma (Lemmas 4.4–4.6)

  Obtain algorithmic access to X_t, Y_t after bucketing on X'_t, Y'_t using Lemma 4.7.

  Update N_S, N_Q, N_{TC} by adding the sample complexity to U_A, query complexity to 1_A, and time complexity, respectively used as part of above three steps.

  if $N_S > B_S$ *or* $N_Q > B_Q$ *or* $N_{TC} > B_{TC}$ **then**

    Abort and return $\perp$

Return Algorithmic access to X_T.

We prove Theorem 4.2 using the following three main arguments, whose intuition we summarize below. As part of these arguments and claims to come, we will require the following definition. For the fixed absolute constant $\eta > 0$ from Theorem 4.3, define the potential Φ as

$$\Phi_{X,Y}(X', Y') := \mathrm{d}[X'; Y'] + \eta\big(\mathrm{d}[X; X'] + \mathrm{d}[Y; Y']\big).$$

1. We first comment on the effects of bucketing as part of each iteration in Algorithm 2. Particularly, we show in Claim 4.10 that bucketing in any iteration t after randomly choosing the GGMT family does not increase the potential too much i.e., $\Phi_{X_{t-1},Y_{t-1}}(X_t, Y_t)$ is close to $\Phi_{X_{t-1},Y_{t-1}}(X'_t, Y'_t)$ which would have been obtained without bucketing. We also show in Claim 4.11 that the cost of algorithmic access to X_t, Y_t remains bounded due to bucketing, in expectation.

2. We next show that in iteration t, whenever the current entropic Ruzsa distance $d_t := \mathrm{d}[X_{t-1}, Y_{t-1};]$ is still above the terminal threshold d_{term}, a random GGMT step followed by bucketing has constant probability of making a constant-factor decrement while restoring constant slack and keeping the local sampling costs under control. We call such iterations *good*. Moreover, we show that Algorithm 2 is *stable*. If the current entropic Ruzsa distance is less than d_{term} then with constant probability, the new entropic Ruzsa distance remains less than d_{term} and thus remain in the terminal regime. This is done in Claim 4.12.

3. We then finally iterate the one-step guarantee in the step above. Since every good iteration decreases the distance geometrically from its initial value at most $\log K$, only $O(\log\log K)$ good iterations are needed to reach constant entropic Ruzsa distance. Particularly, we show that an all-good trajectory reaches the terminal regime, remains there, and has sample/query complexity $\mathrm{poly}(K)$ and time

complexity $O(n \cdot \mathrm{poly}(K))$. Such trajectories occur with probability at least $\Omega(1/\mathrm{polylog}\, K)$. This is shown in Claims 4.12–4.15.

We prove these three claims in more detail below.

4.4.1 Effect of bucketing on the potential

We first show that performing bucketing does not increase the potential for any of the five GGMT families, by too much.

Claim 4.10. *Suppose we have algorithmic access to random variables (X, Y). Define $d := \mathsf{d}[X;Y]$. Fix f to be one of the five families of Theorem 4.3 and let (X', Y') be obtained by applying f to (X, Y). Let $(\widehat{X}, \widehat{Y})$ be the variables obtained after applying bucketing of Lemma 4.7 to (X', Y'). If $\Delta[X], \Delta[Y] \le L_0$ then*

$$\mathop{\mathbb{E}}_{J_X, J_Y}[\Phi_{X,Y}(\widehat{X}, \widehat{Y})] \le \mathop{\mathbb{E}}_{Z}[\Phi_{X,Y}(X', Y')] + 2\log(6L_0 + 12d + 3),$$

where J_X, J_Y are the bucketing labels corresponding to X', Y' respectively and Z is the random variable(s) corresponding to any fiber label of f.

Proof. Note that for f being SelfFiber, CrossFiber, Endgame involves sampling fiber labels as part of implementing X', Y'. We denote these natural labels as z and the corresponding random variable(s) as Z. We now condition on $Z = z$ so that X', Y' are fixed laws, and let J_X, J_Y be their independent bucket labels.

From Ruzsa's triangle inequality (Eq. (6)), we have that $\mathsf{d}[X;X], \mathsf{d}[Y;Y] \le 2d$. We now observe that from Lemmas 4.4–4.6 that (X', Y') obtained after application of any family f from Theorem 4.3 satisfies

$$\mathop{\mathbb{E}}_{z\sim Z}[\Delta[X']] \le 2L_0 + 4d, \quad \text{and} \quad \mathop{\mathbb{E}}_{z\sim Z}[\Delta[Y']] \le 2L_0 + 4d. \tag{38}$$

For SelfSum and CrossSum, the bounds are $\Delta[X'], \Delta[Y'] \le L_0 + 4d$ and $\Delta[X'], \Delta[Y'] \le L_0 + 2d$, respectively (Lemma 4.4). For SelfFiber and CrossFiber, the bound in both cases is $\Delta[X'], \Delta[Y'] = 2L_0$ in expectation (Lemma 4.5). For Endgame, the bounds are $\Delta[X'], \Delta[Y'] \le 2L_0 + 4d$ in expectation (Lemma 4.6). This verifies Eq. (38) for every case of f.

Let us define

$$I_X := I[X' : J_X \mid z], \qquad I_Y := I[Y' : J_Y \mid z].$$

Applying Lemma 3.6 successively to the two bucket labels then gives

$$\begin{aligned}
\mathop{\mathbb{E}}_{J_X, J_Y}[\mathsf{d}[\widehat{X};\widehat{Y}] \mid z] &\le \mathsf{d}[X';Y'] + \frac{1}{2}(I_X + I_Y),\\
\mathop{\mathbb{E}}_{J_X}[\mathsf{d}[X;\widehat{X}] \mid z] &\le \mathsf{d}[X;X'] + \frac{1}{2}I_X,\\
\mathop{\mathbb{E}}_{J_Y}[\mathsf{d}[Y;\widehat{Y}] \mid z] &\le \mathsf{d}[Y;Y'] + \frac{1}{2}I_Y.
\end{aligned} \tag{39}$$

Starting from the definition of Φ, we then obtain

$$\begin{aligned}
\mathop{\mathbb{E}}_{J_X, J_Y}[\Phi_{X,Y}(\widehat{X}, \widehat{Y}) \mid z] &= \mathop{\mathbb{E}}_{J_X, J_Y}[\mathsf{d}[\widehat{X};\widehat{Y}] \mid z] + \eta\Big(\mathop{\mathbb{E}}_{J_X}[\mathsf{d}[X;\widehat{X}] \mid z] + \mathop{\mathbb{E}}_{J_Y}[\mathsf{d}[Y;\widehat{Y}] \mid z]\Big)\\
&\le \Phi_{X,Y}(X', Y') + \frac{1+\eta}{2}(I_X + I_Y),
\end{aligned} \tag{40}$$

where we used Eq. (39) and the definition of Φ in the second line. We now bound the mutual information I_X, I_Y. Starting from Claim 4.9 which is true for any z, we obtain

$$\begin{aligned}\mathop{\mathbb{E}}_{Z}[I_X + I_Y] &\le \mathop{\mathbb{E}}_{Z}[\log(3\Delta[X'] + 3)] + \mathop{\mathbb{E}}_{Z}[\log(3\Delta[Y'] + 3)] \\ &\le \log\left(3\mathop{\mathbb{E}}_{Z}[\Delta[X']] + 3\right) + \log\left(3\mathop{\mathbb{E}}_{Z}[\Delta[Y']] + 3\right) \qquad (41)\\ &\le 2\log(6L_0 + 12d + 3), \qquad (42)\end{aligned}$$

where we have used Jensen's inequality in the second line after noting that the concavity of $\log(\cdot)$ and used Eq. (38) in the final line. Noting that $(1+\eta)/2 \le 1$ and substituting Eq. (42) into Eq. (40) gives us

$$\mathop{\mathbb{E}}_{J_X,J_Y}[\Phi_{X,Y}(\widehat{X},\widehat{Y})] \le \mathop{\mathbb{E}}_{Z}[\Phi_{X,Y}(X',Y')] + 2\log(6L_0 + 12d + 3),$$

which is the desired result. This completes the proof. □

Cost of algorithmic access after bucketing. We first make the notion of cost more precise. In each iteration t of Algorithm 2, we take a pair of independent random variables X_{t-1}, Y_{t-1}, choose a family F from Theorem 4.3, set X'_t, Y'_t to be the pair obtained after applying F and then obtain X_t, Y_t by applying bucketing (Lemma 4.7 and Algorithm 1). We do not store the laws of X_t, Y_t explicitly but rather implement algorithmic access by specifying their corresponding samplers and density coins. These are implemented from the algorithmic access to X'_t, Y'_t as was shown in Lemma 4.7 which in turn is implemented from access to X_{t-1}, Y_{t-1} as was shown in Lemmas 4.4–4.6. As access is implemented from prior level access, the cost of the procedures for X_t, Y_t will then depend on the number of calls to the access procedures of X_{t-1}, Y_{t-1} which we need to control.

Going from X_{t-1}, Y_{t-1} to X'_t, Y'_t, we mainly need for the rejection samplers used for sampling labels of fibers as part of applying F. If the fiber proposal acceptance probability is a, the sampler makes a^{-1} proposals in expectation. Since these overheads multiply when access procedures are composed across iterations, it is convenient to record them logarithmically as $\log(a^{-1})$. We therefore define C_{fib} to be the sum of the logarithms of the reciprocal acceptance probabilities (see Lemmas 4.5, and 4.6 for formal expressions) of the fiber rejection samplers used by the selected family F as follows:

$$C_{\text{fib}} := \begin{cases} 0, & \text{for SelfSum and CrossSum,} \\ \log\dfrac{M_X}{p_{X+X}(z_X)} + \log\dfrac{M_Y}{p_{Y+Y}(z_Y)}, & \text{for SelfFiber with labels } (z_X, z_Y), \\ \log\dfrac{M_Y}{p_{X+Y}(z_0)} + \log\dfrac{M_X}{p_{X+Y}(z_1)}, & \text{for CrossFiber with labels } (z_0, z_1), \\ \log\dfrac{M_Y}{p_{X+Y}(z)} + \log\dfrac{M_X}{p_{X+Y}(w+z)}, & \text{for Endgame with label } (w, z). \end{cases} \qquad (43)$$

For SelfSum and CrossSum, no fiber sampling is needed, and hence $C_{\text{fib}} = 0$.

We now account for the overhead due to bucketing as we go from X'_t, Y'_t to X_t, Y_t. From Claim 4.8, we have that the bucketed sampler uses $O\left(\frac{2^{j+1}}{\pi_{X,j}}\right)$ calls to its parent access procedures (i.e., X'_t, Y'_t) conditional on the bucketing label $J_X = j$, while its density coin uses at most 2^{j+1} parent-coin calls. Since $\pi_{X,j} \le 1$ and $\pi_{Y,j} \le 1$, the former expression controls both access types. This motivates defining the overall logarithmic cost overhead of going from X_{t-1}, Y_{t-1} to X_t, Y_t as

$$C_{\text{access}} := C_{\text{fib}} + \log\frac{2^{J_X+1}}{\pi_{J_X}} + \log\frac{2^{J_Y+1}}{\pi_{J_Y}} \qquad (44)$$

Up to some constant, $2^{C_{\text{access}}}$ then bounds the calls made by access procedures of X_t, Y_t to X_{t-1}, Y_{t-1} in iteration t. We now show that C_{access} remains bounded in expectation over the course of any iteration.

Claim 4.11. *Consider the context of Claim 4.10. We then have*

$$\mathbb{E}[C_{\text{access}}] \le 24(d + L_0 + 1).$$

Proof. Note that for f being SelfFiber, CrossFiber, Endgame involves sampling fiber labels as part of implementing X', Y'. We denote these natural labels as z and the corresponding random variable(s) as Z. We now condition on $Z = z$ so that X', Y' are fixed laws, and let J_X, J_Y be their independent bucket labels. We denote $\pi_{J_X}(j) := \Pr[J_X = j]$ and $\pi_{J_Y}(j) := [J_Y = j]$.

To bound the expression in the claim statement, we will bound the terms separately. From Ruzsa's triangle inequality (Eq. (6)), we have that $\mathsf{d}[X;X], \mathsf{d}[Y;Y] \le 2d$. We now observe that from Lemmas 4.4–4.6 that

$$\mathop{\mathbb{E}}_{z\sim Z}[C_{\text{fib}}] \le 2L_0 + 4d, \tag{45}$$

For SelfSum and CrossSum, the expression is equal to 0 since no fiber sampling is involved (Lemma 4.4). For SelfFiber and CrossFiber, the bounds are $L_0 + 4d$ and $L_0 + 2d$, respectively (Lemma 4.5). For Endgame, the bound is $2L_0 + 2d$ (Lemma 4.6). This verifies Eq. (38) for every case of f.

We now note that

$$\mathop{\mathbb{E}}_{J_X}[J_X + \log(1/\pi_{X,J_X})] \le \Delta[X'] + \mathbf{H}[J_X] \le \Delta[X'] + \log(3\Delta[X'] + 3), \tag{46}$$

where we have used $\mathbb{E}_{J_X}[J_X] \le \Delta[X']$ (Eq.(36) from proof of Claim 4.9) and $\mathbb{E}_{J_X}[\log(1/\pi_{X,J_X})] = \mathbf{H}[J_X]$ since $\pi_{X,j} = \Pr[J_X = j]$ in the second inequality followed by Eq. (37) (from proof of Claim 4.9). Similarly, we can show that

$$\mathop{\mathbb{E}}_{J_Y}[J_Y + \log(1/\pi_{Y,J_Y})] \le \Delta[Y'] + \mathbf{H}[J_Y] \le \Delta[Y'] + \log(3\Delta[Y'] + 3).$$

Combining Eq. (45) and Eq. (46) gives us

$$\begin{aligned}
&\mathop{\mathbb{E}}_{Z,J_X,J_Y}\left[C_{\text{fib}} + \log\frac{2^{J_X}}{\pi_{X,J_X}} + \log\frac{J_Y}{\pi_{Y,J_Y}}\right] \\
&\le (2L_0 + 4d) + \mathop{\mathbb{E}}_{z\sim Z}\left[\Delta[X'] + \Delta[Y']\right] + \mathop{\mathbb{E}}_{z\sim Z}\left[\log(3\Delta[X'] + 3) + \log(3\Delta[Y'] + 3)\right] \\
&\le (2L_0 + 4d) + (4L_0 + 8d) + \log\left(\mathop{\mathbb{E}}_{z\sim Z}[3\Delta[X'] + 3]\right) + \log\left(\mathop{\mathbb{E}}_{z\sim Z}[3\Delta[Y'] + 3]\right) \\
&\le 6L_0 + 12d + 2\log(6L_0 + 12d + 3) && (47) \\
&\le 12L_0 + 24d + 3, && (48)
\end{aligned}$$

where we have used $\mathbb{E}_{z\sim Z}\left[\Delta[X']\right], \mathbb{E}_{z\sim Z}\left[\Delta[Y']\right] \le 2L_0 + 4d$ from Eq. (38) (proof of Claim 4.10) in the third line along with application of Jensen's inequality after noting the concavity of log. We used Eq. (38) again in the fourth line. □

4.4.2 Guarantees of a good iteration

In Algorithm 2, we choose the GGMT family to be applied uniformly at random. We thus do not have the promise that the energy Φ will reduce for all choices made. Moreover, even if Φ reduced, it is possible that the algorithmic access procedures of the resulting variables X_t, Y_t in iteration t require prohibitively many calls to previous access procedures making them expensive to implement. We thus define a *good*

iteration as one where Φ reduces, the slack of resulting X_t, Y_t remain bounded and thereby the cost of their algorithmic access remains bounded. We will make this more formal shortly. Let us quickly comment on the promise of any iteration in Algorithm 2.

We show that an iteration in Algorithm 2 is good with high probability. In particular, we consider the two cases of (i) when the entropic Ruzsa distance $d_{t-1} := \mathrm{d}[X_{t-1}; Y_{t-1}]$ at the beginning of an iteration t is higher than the specified terminal Ruzsa distance d_{term}, and (ii) when $d_{t-1} \le d_{\mathrm{term}}$.

In case (i), we show that Algorithm 2 will choose a pair X_t, Y_t which will reduce the potential Φ and keep the cost of algorithmic access bounded, with high probability. In case (ii), since $d_{t-1} \le d_{\mathrm{term}}$ for $t < T_{\max}$, it is possible that X_t, Y_t are chosen such that the potential increases and we escape the terminal regime. We show that Algorithm 2 is *stable* and that with high probability, X_t, Y_t are chosen such that we remain in the terminal regime. The above two stated results are shown formally in the claim below.

Claim 4.12. *Let $\eta \in (0,1)$ be the promise of Theorem 4.3. Suppose we are in iteration t. Define absolute constants $L_0 = 256/\eta$, $\xi = 768/\eta$ and $d_{\mathrm{term}} = \max\{L_0 + 1, 16/\eta \log(64/\eta)\}$. Suppose that we have algorithmic access to X_{t-1}, Y_{t-1} such that*

$$\Delta[X_{t-1}], \Delta[Y_{t-1}] \le L_0, \qquad d_{t-1} := \mathrm{d}[X_{t-1}; Y_{t-1}].$$

Define $d := \max\{d_{t-1}, d_{\mathrm{term}}\}$. Then, running one iteration of steps 6–8 of Algorithm 2, starting from the current pair X_{t-1}, Y_{t-1} outputs X_t, Y_t with probability $\ge \eta/40$ such that

$$\Phi_{X_{t-1},Y_{t-1}}(X_t, Y_t) \le (1 - \eta/2)d, \quad \Delta[X_t], \Delta[Y_t] \le L_0, \quad C_{\mathrm{access}} \le \xi(d+1).$$

Proof. To simplify the notation, let us denote $X := X_{t-1}$ and $Y := Y_{t-1}$ as the variables at the start of iteration t. Let the outputs from step 6 of Algorithm 2 be $X' := X'_t$ and $Y' := Y_t$, which are obtained before bucketing. Let Z be the random variable(s) denoting any of the natural fiber labels chosen as part of obtaining X', Y'. Let $\widehat{X} := X_t$ and $\widehat{Y} := Y_t$ be the variables obtained after applying bucketing. We will still denote $d_{t-1} := \mathrm{d}[X_{t-1}; Y_{t-1}]$.

By Theorem 4.3, there is a family $F^\star$ such that,

$$\mathop{\mathbb{E}}_{z\sim Z}\left[\Phi_{X,Y}(X', Y') \mid F^\star\right] \le (1-\eta)d_{t-1}.$$

Step 6 of Algorithm 2 chooses $F^\star$ with probability at least $1/5$. From Claim 4.10, we then have that

$$\mathop{\mathbb{E}}_{Z,J_X,J_Y}[\Phi_{X,Y}(\widehat{X}, \widehat{Y}) \mid F^\star] \le (1-\eta)d_{t-1} + 2\log(6L_0 + 12d_{t-1} + 3). \tag{49}$$

Let us analyze the upper bound of the above expression by considering two cases: (i) $d_{t-1} > d_{\mathrm{term}}$, and (ii) $d_{t-1} \le d_{\mathrm{term}}$.

For case (i), we now upper bound the right hand side of Eq. (49). Choosing $d_{\mathrm{term}} = \max\{L_0 + 1, 16/\eta \log(64/\eta)\}$ ensures that

$$2\log(6L_0 + 12d_{t-1} + 3) \le 2\log(18d_{t-1})$$

since $d_{\mathrm{term}} \le d_{t-1}$. To bound $2\log(18d_{t-1})$, we use the following observation. Consider the function $g(x) := 2\log(18x)/x, \forall x > 0$. We note that its derivative is $g'(x) = 2(1 - \ln(18x))/(x^2 \ln 2) < 0$ for $x \ge \exp(1)/18$ and hence $g(x)$ is a decreasing function for $x \ge \exp(1)/18$. Let us define $x_0 := 16/\eta \log(64/\eta)$ which we note is greater than $\exp(1)/18$ for $\eta \le 1$. We then have that

$$\frac{2\log(18d_{t-1})}{d_{t-1}} \le \frac{2\log(18x_0)}{x_0} \le \frac{4\log(64/\eta)}{x_0} = \frac{\eta}{4} \implies 2\log(18d_{t-1}) \le \frac{\eta}{4} d_{t-1},$$

where we used $18x_0 \le (64/\eta)^2$ in the second inequality, and definition of x_0 in the final inequality before the implication. Combining the above bound with Eq. (49) then gives us that

$$\mathbb{E}\left[\Phi_{X,Y}(\widehat{X},\widehat{Y}) \mid F^\star\right] \le \left(1 - \frac{3\eta}{4}\right) d_{t-1}. \tag{50}$$

In case (ii), we can then bound the expression from Eq. (49) as

$$\mathop{\mathbb{E}}_{J_X,J_Y}\left[\Phi_{X,Y}(\widehat{X},\widehat{Y}) \mid F^\star\right] \le (1-\eta)d_{\text{term}} + 2\log(6L_0 + 12d_{\text{term}} + 3).$$

Let us choose $d_{\text{term}} = \max\{L_0 + 1, 16/\eta \log(64/\eta)\}$ as before. This ensures that $2\log(6L_0 + 12d + 3) \le 2\log(18d_{\text{term}})$. Given our choice, we can write $d_{\text{term}} = 16x/\eta$ for some $x \ge \log(64/\eta) \ge 6$ since $\eta < 1$. We note that $6x \le 2^x$ since $x \ge 6$ and hence $1/\eta \le 2^x/64$. This gives us $2\log(18d_{\text{term}}) = 2\log(288x/\eta)$ and $288x/\eta \le 2^{2x}$ from our earlier observations. This overall gives us that $2\log(18d_{\text{term}}) \le 4x = \eta d/4$. We then have

$$\mathbb{E}\left[\Phi_{X,Y}(\widehat{X},\widehat{Y}) \mid F^\star\right] \le \left(1 - \frac{3\eta}{4}\right) d_{\text{term}}. \tag{51}$$

Let us define $d := \max\{d_{t-1}, d_{\text{term}}\}$. We then have for both (i) $d_{t-1} > d_{\text{term}}$ (Eq. (50)) and (ii) $d_{t-1} \le d_{\text{term}}$ (Eq. (51)) that

$$\mathbb{E}\left[\Phi_{X,Y}(\widehat{X},\widehat{Y}) \mid F^\star\right] \le \left(1 - \frac{3\eta}{4}\right) d. \tag{52}$$

Since the potential is nonnegative, Markov's inequality now gives

$$\Pr\left[\Phi_{X,Y}(\widehat{X},\widehat{Y}) \le (1-\eta/2)d \mid F^\star\right] \ge 1 - \frac{\mathbb{E}[\Phi_{X,Y}(\widehat{X},\widehat{Y}) \mid F^\star]}{1-\eta/2} \ge 1 - \frac{1-3\eta/4}{1-\eta/2} = \frac{\eta/4}{1-\eta/2} \ge \eta/4, \tag{53}$$

where we have used Eq. (52) in the second inequality and the fact that $\eta \in (0,1)$ in the final inequality.

Noting that after the natural label $Z = z$ has been fixed, the laws of X', Y' are fixed and we can then apply Claim 4.9 to obtain (which is true regardless of the value of d_{t-1} in comparison with d_{term})

$$\mathop{\mathbb{E}}_{J_X}\left[\Delta[\widehat{X}] \mid Z = z, F^\star\right] < 8, \quad \mathop{\mathbb{E}}_{J_Y}\left[\Delta[\widehat{Y}] \mid Z = z, F^\star\right] < 8.$$

Noting that the above expression is true for all $Z = z$, we then have after adding both expressions that

$$\mathop{\mathbb{E}}_{Z,J_X,J_Y}\left[\Delta[\widehat{X}] + \Delta[\widehat{Y}] \mid F^\star\right] < 16.$$

Applying Markov's inequality again, we obtain

$$\Pr\left[(\Delta[\widehat{X}] + \Delta[\widehat{Y}]) \ge L_0 \mid F^\star\right] \le 16/L_0. \tag{54}$$

Finally, from Claim 4.11 which holds true for any family f, we have that

$$\mathbb{E}[C_{\text{access}} \mid F^\star] \le 24(L_0 + d_{t-1} + 1) \le 48d,$$

where we have used the fact that $d_{\text{term}} \ge L_0 + 1$ and used the definition $d = \max\{d_{t-1}, d_{\text{term}}\}$. Let $\xi > 0$ be a parameter to be determined later. Applying Markov's inequality considering the above expression for C_{access}, we then have that

$$\Pr\left[C_{\text{access}} \ge \xi(d+1) \mid F^\star\right] \le \frac{48d}{\xi(d+1)}. \tag{55}$$

Let us define E_1 as the event that $\Phi_{X,Y}(\widehat{X},\widehat{Y}) \le (1-\eta/2)d$, E_2 as the event that $\Delta[\widehat{X}] + \Delta[\widehat{Y}] \le L_0$, and E_3 as the event that $C_{\text{access}} \le \xi(d+1)$. Let us denote $\overline{E}$ as the complement of the event E. Applying an union bound, we have that

$$\begin{aligned}
&\Pr\left[\cap_{i\in[3]} E_i \mid F^\star\right] \\
&\ge 1 - \sum_{i\in[3]} \Pr\left[\overline{E}_i \mid F^\star\right] \\
&\ge \Pr\left[\Phi_{X,Y}(\widehat{X},\widehat{Y}) \le (1-\eta/2)d \mid F^\star\right] - \Pr\left[(\Delta[\widehat{X}] + \Delta[\widehat{Y}]) \ge L_0 \mid F^\star\right] - \Pr\left[C_{\text{access}} \ge \xi(d+1) \mid F^\star\right] \\
&\ge \frac{\eta}{4} - \frac{16}{L_0} - \frac{48d}{\xi(d+1)},
\end{aligned} \tag{56}$$

where we have used Eqs (53),(54),and (55) in the last inequality. Choosing $L_0 = 256/\eta$ and $\xi = 768/\eta$ ensures that

$$\Pr\left[\cap_{i\in[3]} E_i \mid F^\star\right] \ge \eta/8.$$

Noting that $\cap_{i\in[3]} E_i$ is the desired good iteration and $F^\star$ is chosen uniformly at random (independent of other choices), we then have that

$$\Pr\left[\cap_{i\in[3]} E_i\right] \ge \eta/40.$$

This completes the proof. □

4.4.3 Large fraction of trajectories are good and hit terminal regime

We have so far shown that any iteration in Algorithm 2 is good (Claim 4.12). We now show that there is a large fraction of trajectories (or sequence of iterations) that terminate quickly and where each iteration in the trajectory is promised to be good. Let us define the maximum number of iterations that Algorithm 2 is allowed to run for as

$$T := \min\{t \ge 0 : (1-\eta/2)^t \log K \le d_{\text{term}}\}. \tag{57}$$

Claim 4.13. *Consider the context of Theorem 4.2. Running Algorithm 2 for $T = O(\log\log K)$ iterations ensures that with probability $\ge \Omega(1/\text{polylog}\, K)$, each iteration is good in the sense of Claim 4.12 and*

$$d_T \le d_{\text{term}}, \quad \Delta[X_T], \Delta[Y_T] \le L_0$$

and

$$\sum_{t<T} (\widetilde{d}_t + 1) = O(\log K), \quad \mathrm{d}[U_A; X_T] = O(\log K),$$

where $\widetilde{d}_t := \max\{\mathrm{d}[X_t; Y_t], d_{\text{term}}\}$.

Proof. At $t = 0$, we set the initial random variables X_0 and Y_0 to have the distribution of U_A. The initial entropic Ruzsa distance is then

$$d_0 := \mathrm{d}[X_0; Y_0] = \mathbf{H}[U_A + U_A'] - \mathbf{H}[U_A] \le \log|A+A| - \log|A| \le \log K, \tag{58}$$

where we have used that $(U_A + U_A')$ is supported on $(A+A)$ in the second inequality and that A has doubling constant K in the final inequality. The initial slack is zero from a quick calculation:

$$\Delta[X_0] = \Delta[Y_0] = \mathbf{H}[U_A] + \log M_{U_A} = \log|A| - \log|A| = 0,$$

where we noted that $M_{U_A} = 1/|A|$ by definition. Let G_s be the event that the iteration $s \in [T]_0$ is good in the sense of Claim 4.12 (i.e., decreases the potential, has constant slack, and has bounded cost of algorithmic access). The event $\cap_{s<t} G_s$ then corresponds to all the iterations from 0 up to t being good. Note that at the end of iteration s, the resulting current pair X_s, Y_s satisfies the input requirements to Claim 4.12. We then have from Claim 4.12 that

$$\Pr[G_t \mid \cap_{s<t} G_s] \geq \eta/40 \tag{59}$$

Let us define $p := \eta/40$. Applying the chain rule for conditional probabilities over an entire T-length trajectory then gives us

$$\Pr\left[\cap_{s\in[T]} G_s\right] = \prod_{t=1}^{T} \Pr[G_t \mid \cap_{s<t} G_s] \geq p^T, \tag{60}$$

where we applied Eq. (59) in the second inequality. Let us define $\rho := (1 - \eta/2)$ and recall the definition of $T = \min\{t \geq 0 : \rho^t \log K \leq d_{\text{term}}\}$ from Eq. (57). Suppose that $\log K > d_{\text{term}}$ or else X_0 already satisfies the required guarantees of Theorem 4.2 and we can set $T = 0$. We then have that

$$T \leq \frac{\log\left(\log K/d_{\text{term}}\right)}{\log(1/\rho)} + 1 \leq 1 + \frac{2}{\eta}\log\log K = O\left(\eta^{-1}\log\log K\right), \tag{61}$$

where we used the fact that $d_{\text{term}} \geq 1$ and that $\log(1/\rho) = \log(1/(1-\eta/2)) \geq \eta/2$ since $-\ln(1-x) \geq x, \forall x < 1$. Substituting the above bound on T back into Eq. (60), we then have that the probability of a good trajectory is

$$\Pr\left[\cap_{s\in[T]} G_s\right] \geq p^T \geq p(p)^{2/\eta \log\log K} \geq p(\log K)^{2/\eta \log p} \geq \frac{\eta/40}{(\log K)^{2/\eta \log(40/\eta)}} = \Omega(1/\text{polylog}\, K), \tag{62}$$

where we used $0 < p \leq 1$ and Eq. (61) in the second inequality, the fact that $p \geq \eta/40$ in the fourth inequality and $\eta = O(1)$ in the final inequality.

We now derive the promises on X_T, Y_T obtained at the end of a good trajectory in $\mathcal{G} := \cap_{s\in[T]} G_s$. Claim 4.12 promises us that on every good iteration, we have

$$d_{t+1} \leq \Phi_{X_t,Y_t}(X_{t+1}, Y_{t+1}) \leq \rho \widetilde{d}_t, \tag{63}$$

where $\widetilde{d}_t = \max\{\mathrm{d}[X_t; Y_t], d_{\text{term}}\}$ and noted the definition of Φ for the lower bound. Thus in every iteration t, while $d_t > d_{\text{term}}$, we have $d_{t+1} \leq \rho d_t$. If $\tau < T$ is the first iteration that $d_\tau \leq d_{\text{term}}$, then for a good trajectory in $\mathcal{G}$, $d_T \leq d_{\text{term}}$ due to Eq. (63). If there is no such iteration $\tau < T$, then we have that

$$d_T \leq \rho^T d_0 \leq \rho^T \log K \leq d_{\text{term}},$$

by the definition of T (see Eq. (57)). Thus, we have that for every trajectory in $\mathcal{G}$, $d_T \leq d_{\text{term}}$.

To evaluate $\sum_{t<T}(\widetilde{d}_t + 1)$ for a trajectory in $\mathcal{G}$, we will again define τ to be the first iteration such that $d_\tau \leq d_{\text{term}}$. We then have that

$$\sum_{t<T}(\widetilde{d}_t + 1) \leq \sum_{t<\tau} \rho^t \log K + \sum_{\tau\leq t<T} d_{\text{term}} + T \leq \frac{\log K}{1-\rho} + (d_{\text{term}} + 1)T = O(\log K) \tag{64}$$

where we used Eq. (63) and the fact that $d_0 \leq \log K$ followed by the upper bound on T from Eq. (61) along with the fact that $\eta = O(1)$.

To bound the entropic Ruzsa distance $\mathrm{d}[U_A; X_T]$, we note that $X_0 = U_A$ and use the entropic Ruzsa triangle inequality to obtain

$$\mathrm{d}[U_A; X_T] \leq \sum_{t=1}^{T} d[X_{t-1}; X_t] \leq \frac{\rho}{\eta}\sum_{t=1}^{T} \widetilde{d}_t = O(\log K),$$

where in the second inequality used that the definition of the potential Φ and Claim 4.12 implies $\eta \mathrm{d}[X_{t-1}; X_t] \le \Phi_{X_{t-1},Y_{t-1}}(X_t, Y_t) \le \rho \widetilde{d}_{t-1}$ and then used Eq. (64) in the final inequality. The bound on $\Delta[X_T], \Delta[Y_T]$ follows from the promise of the last good iteration (Claim 4.12) if $T > 0$ or from the fact that $\Delta[X_0] = \Delta[Y_0] = 0$ if $T = 0$. This completes the proof. □

4.4.4 Cost of good trajectories

Claim 4.14. *Consider the context of Theorem 4.2. Let $\mathcal{G}$ be the set of trajectories satisfying the promise of Claim 4.13. Then, Algorithm 2 implements algorithmic access to X_T, Y_T as the end of a length-T trajectory Γ satisfying the promise of Claim 4.13 with cumulative cost of C_S sample complexity to U_A, C_Q query complexity to 1_A, and time complexity C_{TC} such that*

$$\mathbb{E}[C_S \mid \Gamma \in \mathcal{G}] = P_1(K), \quad \mathbb{E}[C_Q \mid \Gamma \in \mathcal{G}] = P_2(K), \text{ and } \mathbb{E}[C_{TC} \mid \Gamma \in \mathcal{G}] = O(n \cdot P_3(K)),$$

where P_1, P_2, P_3 are effective polynomials independent of Γ.

Proof. Let us consider a trajectory Γ and denote F_t to be the selected GGMT family from Theorem 4.3, Z_t to be the tuple of natural fiber/endgame labels required by F_t (or $\emptyset$ tuple for a sum) and let $J_{X,t}, J_{Y,t}$ be the two bucket labels. We will collectively denote the tuple $\gamma_t := (F_t, Z_t, J_{X,t}, J_{Y,t})$. We can then describe the trajectory Γ as

$$\Gamma := \{\gamma_t\}_{t\in[T]} = \{(F_t, Z_t, J_{X,t}, J_{Y,t}\}_{t\in[T]}.$$

Note that with the description of Γ up to iteration t, we can recursively algorithmically access X_t, Y_t even though this is not held explicitly. We also do not record rejected proposals as part of the construction of Γ. We will use $\Gamma_{\le t} := \{\gamma_s\}_{s\in[t]}$ to denote the trajectory up to iteration t.

We now determine the cost of a good trajectory $\Gamma \in \mathcal{G}$ satisfying Claim 4.13. For resource $R \in \{S, Q, TC\}$, let $R(\mathcal{A})$ be the number of samples from U_A, queries to 1_A, or time complexity, respectively, used by one call to an access procedure $\mathcal{A}$ (which is either Sample or Coin). Let $C_R^{(t)}$ be the R-cost of implementing algorithmic access to X_t, Y_t in iteration t (which involves choosing F_t, generating the natural labels Z_t, generating bucket labels $J_{X,t}, J_{Y,t}$, and defining the resulting composed sampler and coin descriptions). Let the per-call cost of accessing the resulting samplers be denoted as $\mathsf{SampleCost}_{R,t}$ and density coins be denoted as $\mathsf{CoinCost}_{R,t}$, which we define for a fixed supported trajectory $\Gamma_{\le t} = G_{\le t}$ as

$$\mathsf{SampleCost}_{R,t}(G_{\le t}) := \max_{V\in\{X_t,Y_t\}} \max_{v\in\mathrm{supp}(V)} \mathbb{E}\left[R(\mathsf{Sample}_V) \,\middle|\, \Gamma_{\le t} = G_{\le t}, \mathsf{Sample}_V = v\right], \tag{65}$$

$$\mathsf{CoinCost}_{R,t}(G_{\le t}) := \max_{V\in\{X_t,Y_t\}} \sup_{x\in\mathbb{F}_2^n} \max_{b\in\mathrm{supp}(\mathsf{Coin}_V(x))} \mathbb{E}[R(\mathsf{Coin}_V(x)) \mid \Gamma_{\le t} = G_{\le t}, \mathsf{Coin}_V(x) = b]. \tag{66}$$

We will also define

$$\mathsf{Cost}_{R,t}(G_{\le t}) := \max\left\{\mathsf{SampleCost}_{R,t}(G_{\le t}), \mathsf{CoinCost}_{R,t}(G_{\le t})\right\} \tag{67}$$

as the maximum cost over one call to any sampler or coin available at iteration t. Note that this maximizes over the two laws X_t, Y_t, the type of access, every coin input, and every supported returned sample or bit. The expectation still averages over the internal randomness of that call. For a depth-t access call $\mathcal{A}$, let O be its outputs. Then we have that

$$\begin{aligned}
\mathbb{E}[R(\mathcal{A}) \mid \Gamma_{\le t} = G_{\le t}] &= \sum_o \Pr(O = o \mid \Gamma_{\le t} = G_{\le t}) \cdot \mathbb{E}[R(\mathcal{A}) \mid \Gamma_{\le t} = G_{\le t}, O = o] \\
&\le \max_o \mathbb{E}[R(\mathcal{A}) \mid \Gamma_{\le t} = G_{\le t}, O = o] \\
&\le \mathsf{Cost}_{R,t}(G_{\le t}).
\end{aligned} \tag{68}$$

Fix a supported good trajectory $G \in \mathcal{G}$, and let $C_t(G)$ be the value of $C_{\text{access},t}$ from Claim 4.11 in iteration $t+1$. We now describe the effect of the involved rejection samplers. Suppose one component is a rejection sampler with one-trial acceptance probability a, accepting trial M, and returned sample Out. If $r_v := \Pr(\text{one trial accepts and returns } v)$, then

$$\Pr(M = m \mid \text{Out} = v) = \frac{(1-a)^{m-1} r_v}{r_v/a} = a(1-a)^{m-1}, \quad \mathbb{E}[M \mid \text{Out} = v] = a^{-1}.$$

Thus, if one trial has conditional expected R-cost at most c for every outcome of its lower-level calls, then the tower property gives

$$\mathbb{E}[R\text{-cost of the rejection sampler} \mid \text{Out} = v] \le \frac{c}{a}.$$

This is the only reason that output conditioning is mentioned: it shows that the usual reciprocal-acceptance multiplier a^{-1} remains valid for $\text{Cost}_{R,t+1}$.

We now bound $\text{Cost}_{R,t}$. Recall that algorithmic access procedures of Sample or Coin for X_t, Y_t at iteration t are built by calling procedures from iteration $t-1$. At the root, $X_0 = Y_0 = U_A$. A sampler call makes one call to U_A, a density-coin call makes one membership query to 1_A, and either call requires $O(n)$ time complexity. Hence

$$\text{Cost}_{S,0} \le 1, \qquad \text{Cost}_{Q,0} \le 1, \qquad \text{Cost}_{T,0} = O(n). \tag{69}$$

Let us now fix a supported good trajectory $\Gamma = G \in \mathcal{G}$ and let $C_t(G)$ be the value of $C_{\text{access},t}$ from Claim 4.11 for iteration t. Sum operations use only constantly many parent calls. A fiber sampler with one-trial acceptance probability a uses a geometric number of trials of mean a^{-1}. Noting that $\log a^{-1} \le C_{\text{fib}}$ (defined in Eq. (44), and hence $a^{-1} \le 2^{C_{\text{fib}}} \le 2^{C_t(G)}$.

For a bucket label j, with probability π_j, Claim 4.8 gives a sample overhead of $O(2^{j+1}/\pi_j)$ and overall uses at most 2^{j+1} parent-coin calls. Note that the logarithms of all these multipliers for the two output laws X_t, Y_t are included in $C_t(G)$. Therefore, the number of calls to algorithmic access procedures at iteration-t by access procedures at iteration-$(t+1)$ is $D \cdot 2^{C_t(G)}$ for an absolute constant $D \ge 1$. We then have

$$\begin{aligned}
\text{Cost}_{S,t+1}(G_{\le t+1}) &\le D \cdot 2^{C_t(G)} \big(\text{Cost}_{S,t}(G_{\le t}) + 1\big), \\
\text{Cost}_{Q,t+1}(G_{\le t+1}) &\le D \cdot 2^{C_t(G)} \big(\text{Cost}_{Q,t}(G_{\le t}) + 1\big), \\
\text{Cost}_{TC,t+1}(G_{\le t+1}) &\le D \cdot 2^{C_t(G)} \big(\text{Cost}_{TC,t}(G_{\le t}) + O(n)\big).
\end{aligned} \tag{70}$$

Since $\Gamma = G \in \mathcal{G}$, we have from Claim 4.12 and Claim 4.13 that

$$\sum_{t<T} C_t(G) \le \xi \sum_{t<T} (\widetilde{d}_t + 1) = O(\log K) \tag{71}$$

Iterating Eq. (70) with initial condition of Eq. (69) gives

$$\text{Cost}_{S,t}(G_{\le t}), \text{Cost}_{Q,t}(G_{\le t}) \le (2D)^t 2^{\sum_{s<t} C_s(G)}, \quad \text{Cost}_{TC,t}(G_{\le t}) \le O(n)(2D)^t 2^{\sum_{s<t} C_s(G)}. \tag{72}$$

Used the fact that $2^{\sum_{s<T} C_s(G)} = K^{O(1)}$ from Eq. (71) and $(2D)^T = (\log K)^{O(1)}$ from Claim 4.13 gives us

$$\text{Cost}_{S,T}(G) = \text{poly}(K), \quad \text{Cost}_{Q,T}(G) = \text{poly}(K), \quad \text{Cost}_{TC,T}(G) = O(n \cdot \text{poly}(K)) \ \forall G \in \mathcal{G}.$$

From Eq. (68), we also have this applies to the cost not conditioned on the returned outcomes.

We have so far bounded the cost of one call to the implemented algorithmic access procedures in any iteration $t \in [T]$. We now bound the cumulative cost C_R over the T rounds to generate all the natural/bucket

labels and thereby implement the required access procedures as part of the trajectory. Let $C_R^{(t)}$ be the portion of C_R spent in iteration t i.e., $C_R = \sum_{t<T} C_R^{(t)}$. Generating the natural labels in iteration $t+1$ require only constantly many calls to the access procedures from iteration t. Conditional on a fixed bucket label j, its label experiment uses fewer than 2^{j+1} current coin calls. Those multipliers are again bounded by $2^{C_t(G)}$. Hence, for an absolute constant $D_1 \geq 1$,

$$\begin{aligned}
\mathbb{E}[C_S^{(t+1)} \mid \Gamma = G] &\leq D_1 2^{C_t(G)} \big(\mathrm{Cost}_{S,t}(G_{\leq t}) + 1\big), \\
\mathbb{E}[C_Q^{(t+1)} \mid \Gamma = G] &\leq D_1 2^{C_t(G)} \big(\mathrm{Cost}_{Q,t}(G_{\leq t}) + 1\big), \\
\mathbb{E}[C_{TC}^{(t+1)} \mid \Gamma = G] &\leq D_1 2^{C_t(G)} \big(\mathrm{Cost}_{TC,t}(G_{\leq t}) + O(n)\big).
\end{aligned}$$

Substituting Eq. (72) into the above equation and then summing over $t < T$ gives us

$$\begin{aligned}
\mathbb{E}[C_S \mid \Gamma = G],\ \mathbb{E}[C_Q \mid \Gamma = G] &\leq D_2 T (2D)^T 2^{\sum_{s<T} C_s(G)} = \mathrm{poly}(K), \\
\mathbb{E}[C_{TC} \mid \Gamma = G] &\leq O(n) D_2 T (2D)^T 2^{\sum_{s<T} C_s(G)} = O(n \cdot \mathrm{poly}(K)),
\end{aligned} \tag{73}$$

for an absolute constant $D_2 \geq 1$. Finally, averaging over the conditional distribution of the good trajectory as

$$\mathbb{E}[C_R \mid \Gamma \in \mathcal{G}] = \mathop{\mathbb{E}}_{G \sim (\Gamma \mid \Gamma \in \mathcal{G})} [\mathbb{E}[C_R \mid \Gamma = G]]$$

gives us

$$\mathbb{E}[C_S \mid \Gamma \in \mathcal{G}] = P_1(K), \quad \mathbb{E}[C_Q \mid \Gamma \in \mathcal{G}] = P_2(K), \quad \mathbb{E}[C_{TC} \mid \Gamma \in \mathcal{G}] = O(n \cdot P_3(K)),$$

for some polynomials P_1, P_2, P_3. This completes the proof. □

In the previous claim, we bounded the expected complexities for trajectories executed by Algorithm 2 conditioned on them being good as per Claim 4.13. We now show that by imposing a complexity budget on Algorithm 2, we can still obtain an algorithm that succeeds with high probability.

Claim 4.15. *Consider the context of Theorem 4.2. There exists a complexity budget B that can be imposed on Algorithm 2 to ensure that it always uses $\mathrm{poly}(K)$ samples from U_A, $\mathrm{poly}(K)$ membership queries to 1_A, and $O(n\,\mathrm{poly}(K))$ time complexity such that with probability $\Omega(1/\mathrm{polylog}\,K)$, the algorithm does not abort and returns exact algorithmic access to X_T satisfying*

$$\mathrm{d}[X_T; X_T] \leq 2d_{\mathrm{term}} = O(1), \quad \mathrm{d}[U_A; X_T] = O(\log K), \quad \Delta[X_T] \leq L_0 = O(1).$$

On this event, each later call to Sample_{X_T} or Coin_{X_T} has expected sample complexity $\mathrm{poly}(K)$, membership-query query complexity $\mathrm{poly}(K)$, and time complexity $O(n \cdot \mathrm{poly}(K))$.

Proof. We will re-use the notation from Claim 4.14 and its proof. Suppose we run Algorithm 2 for T many iterations where T is fixed according to Claim 4.13. We then have that the cumulative complexities C_S, C_Q, C_{TC} for generating a T-length trajectory (Claim 4.14) satisfies

$$\mathbb{E}[C_S \mid \Gamma \in \mathcal{G}] \leq P_1(K), \qquad \mathbb{E}[C_Q \mid \Gamma \in \mathcal{G}] \leq P_2(K), \qquad \mathbb{E}[C_{TC} \mid \Gamma \in \mathcal{G}] \leq P_3(n, K). \tag{74}$$

Let us define the following budgets B_S, B_Q, B_{TC} corresponding to the sample complexity, query complexity and time complexity, respectively

$$B_S := \lceil 6P_1(K) \rceil, \qquad B_Q := \lceil 6P_2(K) \rceil, \qquad B_{T_C} := \lceil 6P_3(n, K) \rceil. \tag{75}$$

In Algorithm 2, we thus define three counters from the start of iteration 1 till the end of iteration T corresponding to each resource and update them over the course of an iteration and trajectory. If an operation is to be taken that would make any counter exceed the corresponding budget as defined above, Algorithm 2 aborts and returns $\perp$.

We now comment on those trajectories whose complexities would not exceed the defined budgets of Eq. (75) and satisfies Claim 4.13. Using the conditional Markov inequality, we have that for every $R \in \{S, Q, TC\}$,

$$\Pr[C_R > B_R \mid \Gamma \in \mathcal{G}] \le \frac{\mathbb{E}[C_R \mid \Gamma \in \mathcal{G}]}{B_R} \le \frac{1}{6}. \tag{76}$$

By an union bound, we then have that

$$\Pr\left[C_S > B_S \text{ or } C_Q > B_Q \text{ or } C_{T_C} > B_{T_C} \mid \Gamma \in \mathcal{G}\right] \le \frac{1}{2}.$$

We then obtain that

$$\Pr\left[\Gamma \in \mathcal{G} \text{ and no abort}\right] = \Pr[\Gamma \in \mathcal{G}] \cdot \Pr[\text{no abort} \mid \Gamma \in \mathcal{G}] \ge \frac{1}{2}\Omega(1/\operatorname{polylog} K) = \Omega(1/\operatorname{polylog} K),$$

where we used Claim 4.13. Note that on this event, the complexities C_R of Algorithm 2 will be within the budgets defined in Claim 4.14. Since $\Gamma \in \mathcal{G}$, we have from Claim 4.13 that

$$\mathrm{d}[X_T; Y_T] \le d_{\text{term}}, \quad \Delta[X_T] \le L_0, \quad \text{and} \quad \mathrm{d}[U_A; X_T] = O(\log K).$$

Finally, applying the entropic Ruzsa triangle inequality gives us

$$\mathrm{d}[X_T; X_T] \le 2\mathrm{d}[X_T; Y_T] \le 2d_{\text{term}}.$$

This proves the desired results of X_T. The exact returned access and its expected per-call resource bounds follow from Claim 4.14. This proves the overall claim. □

4.5 Putting everything together

We are now ready to prove Theorem 4.2.

Proof of Theorem 4.2. We use Algorithm 2 with initialized variables of $X_0 = Y_0 = U_A$. The sample access for each is simply U_A and the density coin for each is queries to 1_A. We note that the slack $\Delta[X_0] = \Delta[Y_0] = 0$ and $d_0 := \mathrm{d}[X_0; Y_0] \le \log K$. We run Algorithm 2 for $T = O(\log \log K)$ many iterations as in Claim 4.13 and under the budgeted complexity of Claim 4.15.

From Claim 4.15, we that have the promise that with probability $\Omega(1/\operatorname{polylog} K)$, the algorithm outputs X_T along with exact algorithmic access to it, satisfying

$$\mathrm{d}[X_T; X_T] = O(1), \qquad \mathrm{d}[U_A; X_T] = O(\log K), \qquad \Delta[X_T] = O(1).$$

Applying the entropic PFR theorem (Theorem 3.5) to X_T then gives us the promise that there exists a subspace $H \le \mathbb{F}_2^n$ such that

$$d[X_T; U_H] = O(d[X_T; X_T]) = O(1)$$

The algorithm uses the following complexity (Claim 4.15): $\operatorname{poly}(K)$ samples from U_A, $\operatorname{poly}(K)$ membership queries to 1_A, and $O(n \cdot \operatorname{poly}(K))$ time. From Claim 4.13, we have that the algorithm implements exact algorithmic access to X_T and the complexity of any call to $\mathsf{Sample}_{X_T}, \mathsf{Coin}_{X_T}$ obey the same bounds as of the overall algorithm (Claim 4.14). This completes the proof. □

5 Extracting a PFR subspace

In Theorem 4.2 of Section 4, we showed how using sample and query access to A which is a set with doubling constant K, we could output algorithmic access (Definition 4.1) to a random variable X for which there exists some subspace $H \leq \mathbb{F}_2^n$ such that

$$\mathrm{d}[X;U_H] = O(1), \quad \mathrm{d}[U_A;X] = O(\log K), \quad \text{and} \quad \Delta[X] = O(1).$$

From the entropic Ruzsa triangle inequality, we also thus have $\mathrm{d}[X;X] = O(1)$. Towards proving Theorem 1.2, our goal is to now learn the the PFR subspace H for A. We do this in two steps:

1. Section 5.1 takes X such that $\mathrm{d}[X;X] = O(1)$ and outputs a basis for a subspace $V \leq \mathbb{F}_2^n$ such that $\mathrm{d}[X;U_V] = O(1)$ and therefore $\mathrm{d}[U_A;U_V] = O(\log K)$.

2. Since the learned V may be larger than A, in Section 5.2, we show how to pass to a suitable V' with $V' \leq |V$ and $|V'| \leq |A|$. Using samples and membership queries to A, we find a translate $a + V'$ and verify if $|A \cap (a+V')|/|A| \geq K^{-O(1)}$.

5.1 Extracting a candidate subspace

Taking the span of samples from X is too crude: even when $\mathrm{d}[X;X] = O(1)$, that span can be much larger than A. We instead pass to the self-sum $X + X'$, retain only sums of sufficiently high probability, and span those heavy self-sums. The self-sum also removes irrelevant affine translations in the support of X. The entropic PFR theorem gives a subspace W_0 with $\mathrm{d}[X;U_{W_0}] = O(1)$, so all but a sufficiently small constant fraction of the mass of X lies in $O(1)$ cosets of W_0. It turns out that enlarging W_0 by their coset directions gives an auxiliary subspace W that contains every sufficiently heavy self-sum. These heavy self-sums also carry a constant fraction of the mass of $X + X'$. The span V of $O(n)$ retained heavy self-sums therefore has constant index in W. By index, we mean that $[W : V] := |W|/|V| = O(1)$. It satisfies $\mathrm{d}[X;U_V] = O(1)$, so the triangle inequality gives $\mathrm{d}[U_A;U_V] = O(\log K)$. Additionally, the bound on the slack of X helps in controlling how efficiently heavy self-sums can be detected. We now formally state the terminal subspace extraction result below for which we will use Algorithm 3.

Lemma 5.1. *Let $C \geq 0$ be a known absolute constant. Suppose that we are given algorithmic access to a random variable X satisfying $\mathrm{d}[X;X] + \Delta[X] \leq C$. Then, there is an algorithm that, with probability at least 0.99, outputs a basis for a linear subspace $V \leq \mathbb{F}_2^n$ satisfying $\mathrm{d}[X;U_V] = O(1)$. The algorithm makes $O(n \log n)$ calls to $\mathsf{Sample}_X, \mathsf{Coin}_X$ and runs in $O(n^3)$ time.*

Algorithm 3: Terminal subspace extraction

Input: Algorithmic access to X (Definition 4.1) satisfying $\mathrm{d}[X;X] + \Delta[X] \le C$ where $C \ge 0$ is a known absolute constant.

Output: A basis for a linear subspace $V \le \mathbb{F}_2^n$ satisfying $\mathrm{d}[X;U_V] = O(1)$ with probability at least 0.99.

1 Set parameters $\tau := 2^{-2(C+1)}$, $m_1 = \lceil 16(n+8) \rceil$ and $m_2 = \left\lceil 48\tau^{-2} \ln(400 m_1) \right\rceil$.

2 Use Lemma 4.4 to construct algorithmic access to $Z = X + X'$, where X' is an independent copy of X.

3 Draw $z_1, \dots, z_{m_1}$ independently from Sample_Z. For each i, let $\widehat{r}_i$ be the empirical mean of m_2 independent calls to $\mathrm{Coin}_Z(z_i)$, and retain z_i if $\widehat{r}_i \ge \tau/2$.

4 **Return** A basis for the span of the retained vectors.

Before proving Lemma 5.1, we first comment on the structural properties of the heavy self-sums.

Claim 5.2. *Fix an absolute constant $C \ge 0$. Suppose that X has envelope M_X and $\mathrm{d}[X;X] + \Delta[X] \le C$. For $Z = X + X'$, where X' is an independent copy of X, set*

$$\tau := 2^{-2(C+1)}, \qquad r(z) := \frac{p_Z(z)}{M_X}, \qquad V_0 := \mathrm{Span}\{z : r(z) \ge \tau/4\}.$$

Then, $\Pr[r(Z) \ge \tau] \ge 1/2$. Moreover, every subspace $V \le V_0$ with $|V| \ge 1/(8M_X)$ satisfies $\mathrm{d}[X;U_V] = O(1)$.

Proof. For every z, the envelope property gives $p_Z(z) = \sum_x p_X(x) p_X(x+z) \le M_X \sum_x p_X(x) = M_X$. Hence $\log(1/r(Z))$ is nonnegative and has expectation $\mathbf{H}[Z] + \log M_X = \mathrm{d}[X;X] + \Delta[X] \le C$. Markov's inequality gives $\Pr[r(Z) < \tau] = \Pr[\log(1/r(Z)) > 2(C+1)] \le C/(2(C+1)) < 1/2$.

By Theorem 3.5, there exists a a subspace W_0 such that $\mathrm{d}[X;U_{W_0}] = O(1)$. Let π be the quotient map modulo W_0 and put $Q := \pi(X)$. Contraction (Lemma 3.4) gives $\mathbf{H}[Q]/2 = \mathrm{d}[Q;0] \le \mathrm{d}[X;U_{W_0}] = O(1)$. Applying the entropy-tail bound in Lemma 3.4 with $\varepsilon := \tau/16$, choose a set S such that $\Pr[Q \in S] \ge 1 - \varepsilon$ and $|S| \le 2^{\mathbf{H}[Q]/\varepsilon} = O(1)$. Set $W := \pi^{-1}(\mathrm{Span}(S+S))$. Since $W/W_0 = \mathrm{Span}(S+S)$ and $|S+S| \le |S|^2$,

$$[W : W_0] = |\mathrm{Span}(S+S)| \le 2^{|S+S|} \le 2^{|S|^2} = O(1).$$

The triangle inequality and the nested-subspace identity in Lemma 3.4 now give

$$\mathrm{d}[X;U_W] \le \mathrm{d}[X;U_{W_0}] + \mathrm{d}[U_{W_0};U_W] = \mathrm{d}[X;U_{W_0}] + \frac{1}{2}\log[W : W_0] = O(1).$$

The entropy-difference bound in the same lemma states that $|\mathbf{H}[X] - \mathbf{H}[U_W]| \le 2\mathrm{d}[X;U_W]$. Since $\mathbf{H}[U_W] = \log|W|$, it follows that

$$\log(M_X|W|) = \Delta[X] + \mathbf{H}[U_W] - \mathbf{H}[X] \le C + 2\mathrm{d}[X;U_W] = O(1).$$

We next show that every self-sum at the cutoff defining V_0 lies in W. This will make any sufficiently large $V \le V_0$ have constant index in W. Let $E := \{x : \pi(x) \notin S\}$. If $z \notin W$, then at least one of x and $x+z$ lies in E for every x: otherwise $\pi(z) = \pi(x) + \pi(x+z) \in S + S \subseteq \mathrm{Span}(S+S)$. Therefore

$$\begin{aligned} p_Z(z) &\le \sum_{x \in E} p_X(x) p_X(x+z) + \sum_{x : x+z \in E} p_X(x) p_X(x+z) \\ &\le 2M_X \Pr[X \in E] \le \frac{\tau}{8} M_X. \end{aligned}$$

Thus every z with $r(z) \geq \tau/4$ lies in W, and hence $V_0 \leq W$.

If $V \leq V_0$ and $|V| \geq 1/(8M_X)$, then

$$[W : V] \leq 8M_X|W| = O(1), \qquad \mathrm{d}[X; U_V] \leq \mathrm{d}[X; U_W] + \frac{1}{2}\log[W : V] = O(1),$$

as required. This completes the proof. □

Proof of Lemma 5.1. Let $C \geq 0$ be the absolute constant such that $\mathrm{d}[X; X] + \Delta[X] \leq C$ and take $\tau := 2^{-2(C+1)}$ as in Claim 5.2. We also set the constants $m_1 := \lceil 16(n+8) \rceil$ and $m_2 = \lceil 48\tau^{-2} \ln(400m_1) \rceil$. Let r, V_0 be as in Claim 5.2 and let us define $H := \{z : r(z) \geq \tau\}$. Conditional on z_i, the estimate $\widehat{r_i}$ is the mean of m_2 Bernoulli variables of mean $r(z_i)$. Hence the additive Chernoff bound and a union bound give

$$\Pr\left[\max_{i \in [m_1]} |\widehat{r_i} - r(z_i)| > \frac{\tau}{4}\right] \leq 2m_1 \exp\left(-\frac{m_2\tau^2}{48}\right) \leq 0.005,$$

where the last inequality follows from our choices of m_1, m_2. On the complementary event, every proposal in $\mathcal{H}$ is retained and every retained proposal lies in V_0. For $i \in [m_1]$, let V_i be the span of the proposals in H among $z_1, \ldots, z_i$. If V is the output, then $V_{m_1} \leq V \leq V_0$.

For $i \in [m_1 - 1]$, while $|V_i| < 1/(8M_X)$, independence of the next proposal and the envelope bound give

$$\Pr[z_{i+1} \in H \setminus V_i \mid z_1, \ldots, z_i] \geq \Pr[Z \in H] - \Pr[Z \in V_i] \geq \frac{1}{2} - M_X|V_i| > \frac{3}{8}.$$

Every such proposal increases $\dim V_i$. Since $M_X \geq \|p_X\|_\infty \geq 2^{-n}$, at most n increases are needed to reach $|V_i| \geq 1/(8M_X)$. Binomial domination and a Chernoff bound therefore give $\Pr\left[|V_{m_1}| \geq 1/(8M_X)\right] \geq 0.995$.

On the intersection of these events, $V \leq V_0$ and $|V| \geq |V_{m_1}| \geq 1/(8M_X)$. The second conclusion of Claim 5.2 therefore gives $\mathrm{d}[X; U_V] = O(1)$, and the union bound gives success probability at least 0.99.

The complexity of Algorithm 3 is as follows. By Lemma 4.4, we have that each call to Sample_Z uses 2 calls to Sample_X, and each call to Coin_Z uses one call to Sample_X along with one call to Sample_X. Since we make m_1 calls to Sample_Z and m_1m_2 calls to Coin_Z in Algorithm 3, overall we make $O(m_1m_2) = O(n \log n)$ many calls to $\mathsf{Sample}_X, \mathsf{Coin}_X$. The generation of proposals and estimation of densities $\widehat{r_i}$ consume $O(n)$ time complexity but the main contribution is in computing the output basis over $O(n)$ retained vectors of length n, which consumes $O(n^3)$ time using Gaussian elimination. This completes the proof. □

5.2 Verifying a good PFR subspace

We now have a linear subspace V that is close to a good terminal distribution X in entropic Ruzsa distance. A good GGMT trajectory also guarantees that X is within $O(\log K)$ of U_A, so by triangle inequality we have

$$\mathrm{d}[U_A; U_V] \leq O(\log K).$$

Since $\mathrm{d}[U_A; U_V] \leq O(\log K)$, the distribution of the V-coset of a uniform point of A has small entropy, so a noticeable fraction $(1/\mathrm{poly}(K))$ of A lies in a coset of V. However, $\mathrm{d}[U_A; U_V] \leq O(\log K)$ does not guarantee the $|V| \leq |A|$ promise of Theorem 1.2. Further, the algorithm chooses a good trajectory with probability $1/\mathrm{polylog}\, K$, so to amplify the success probability via repeated trials, we need a way to detect whether the candidate V obtained is indeed a good PFR subspace for A. To finish the algorithm, we do the following:

1. We pick an arbitrary subspace $V' \leq V$ of codimension $O(\log K)$ such that either $V' = \{0\}$ or $|V'| \leq |A|/2$. Now we have a small enough subspace V'. Further, the heavy V-coset splits into at most $\mathrm{poly}(K)$ cosets of V', so there is a heavy V'-coset. We then observe that a random point $a \sim U_A$ is likely to lie in such a heavy V'-coset with constant probability.

2. It remains only to recognize when this has happened. To this end, for a candidate $a + V'$, we estimate

$$\alpha := \frac{|A \cap (a + V')|}{|A|} \qquad \text{and} \qquad \beta := \frac{|A \cap (a + V')|}{|V'|}.$$

The first quantity checks that $a + V'$ captures a noticeable fraction of A, while the relation $\frac{\beta}{\alpha} = \frac{|A|}{|V'|}$ certifies that V' is not larger than A. Both quantities can be estimated with $K^{O(1)}$ samples and membership queries, yielding an efficiently verifiable PFR candidate. We make this rigorous below.

Lemma 5.3. *Let $A \subseteq \mathbb{F}_2^n$ be nonempty, let $R \geq 1$, and let $0 < \delta \leq 1/2$. Given uniform-sample and membership-oracle access to A and a basis for a subspace $V \leq \mathbb{F}_2^n$, there is an algorithm that either rejects or accepts and outputs an affine coset $a + V'$ with $V' \leq V$ and $\dim V' = \dim V - O(R)$. It has the following guarantees.*

- ***Completeness.*** *If $\mathrm{d}[U_A; U_V] \leq R$, then the algorithm accepts with probability $\Omega(1)$.*
- ***Soundness.*** *Except with probability δ, every accepted output satisfies*

$$|V'| \leq |A|, \qquad \frac{|A \cap (a + V')|}{|A|} \geq 2^{-O(R)}, \qquad \frac{|A \cap (a + V')|}{|V'|} \geq 2^{-O(R)}.$$

The algorithm uses $2^{O(R)} \operatorname{poly}(n, \log(1/\delta))$ time, samples, and membership queries.

Claim 5.4. *Let $A \subseteq \mathbb{F}_2^n$ be nonempty, let $R \geq 1$, and suppose that we are given a sampler for U_A and a basis for a linear subspace $V \leq \mathbb{F}_2^n$ satisfying $\mathrm{d}[U_A; U_V] \leq R$. Then, there is an algorithm that, with probability $\Omega(1)$, outputs an affine coset $a + V'$, where $a \sim U_A$ and $V' \leq V$ satisfies $\dim V' = \dim V - O(R)$ and either $V' = \{0\}$ or $|V'| \leq |A|/2$, such that*

$$\frac{|A \cap (a + V')|}{|A|} \geq 2^{-O(R)}.$$

Further the algorithm uses $\operatorname{poly}(n)$ time and one sample from U_A.

Proof. Let Q denote the random variable corresponding to the coset of V containing a uniform sample from A, that is, $Q = U_A + V$. If $a \sim U_A$ and $v \sim U_V$, then $a + v$ conditioned on $Q = a + V$ is uniform on this coset of V. Therefore, by the entropy chain rule, we have that

$$\begin{aligned} \mathrm{d}[U_A; U_V] &= \mathbf{H}[U_A + U_V] - \frac{1}{2}(\mathbf{H}[U_A] + \mathbf{H}[U_V]) \\ &= \mathbf{H}[Q] + \mathbf{H}[U_V] - \frac{1}{2}\log(|A||V|) = \mathbf{H}[Q] + \frac{1}{2}\log\frac{|V|}{|A|}. \end{aligned}$$

Therefore, $|V| \leq 2^{2R}|A|$. Further, $\mathbf{H}[U_A] = \mathbf{H}[U_A + U_V | U_V] \leq \mathbf{H}[U_A + U_V] = \mathbf{H}[Q] + \log|V|$, so the displayed identity also gives $\mathbf{H}[Q] \leq 2R$. Let $V' \leq V$ be any subspace of codimension $r := \min\{\dim V, \lceil 2R \rceil + 1\}$. Then either $V' = \{0\}$ or $|V'| \leq |A|/2$. If Q' denotes the V'-coset containing a uniform sample from A, then Q is determined by Q' and each V-coset contains $[V : V'] = 2^r$ cosets of V'. Hence $\mathbf{H}[Q'] \leq \mathbf{H}[Q] + r = O(R)$. Draw one sample $a \sim U_A$ and output $a + V'$. Its U_A-mass is $p_{Q'}(Q') = |A \cap (a + V')|/|A|$. Since $\mathbb{E}[\log(1/p_{Q'}(Q'))] = \mathbf{H}[Q'] = O(R)$, Markov's inequality shows that this mass is at least $2^{-O(R)}$ with probability $\Omega(1)$. □

Claim 5.5 (Verifying a candidate subspace)**.** *Let $A \subseteq \mathbb{F}_2^n$, and let $0 < \eta \leq 1$ and $0 < \delta < 1$. There is an algorithm that, given membership-oracle and uniform-sample access to A and an affine representation of a candidate affine subspace $a + V'$, satisfies the following guarantees:*

- ***Completeness.*** *The algorithm accepts with probability at least* $1-\delta$ *if*

$$(V' = \{0\} \text{ or } |V'| \le |A|/2) \quad \text{and} \quad \frac{|A \cap (a+V')|}{|A|} \ge \eta.$$

- ***Soundness.*** *The algorithm rejects with probability at least* $1-\delta$ *if* $a+V'$ *does not satisfy*

$$|V'| \le |A|, \qquad \frac{|A \cap (a+V')|}{|A|} = \Omega(\eta), \qquad \frac{|A \cap (a+V')|}{|V'|} = \Omega(\eta).$$

The algorithm uses $\mathrm{poly}(n, 1/\eta, \log(1/\delta))$ *time, samples, and membership queries.*

Proof. Given its affine representation, we can implement membership-oracle access as well as sample access for $a+V'$ in time $\mathrm{poly}(n)$. Using $m = \mathrm{poly}(1/\eta, \log(1/\delta))$ independent samples from U_A and $U_{a+V'}$, we estimate the quantities

$$\alpha := \frac{|A \cap (a+V')|}{|A|} = \Pr_{x \sim U_A}[x \in a+V'], \quad \text{and} \quad \beta := \frac{|A \cap (a+V')|}{|V'|} = \Pr_{y \sim U_{a+V'}}[y \in A]$$

to get empirical estimates $\widehat{\alpha}$ and $\widehat{\beta}$. If $V' = \{0\}$, we accept exactly when $\widehat{\alpha} \ge \eta/2$. Otherwise, we accept exactly when $\widehat{\alpha} \ge \eta/2$ and $\widehat{\beta} \ge 3\widehat{\alpha}/2$. Completeness and soundness of this test can be argued as follows. By Chernoff's inequality and a union bound, with probability at least $1-\delta$, both estimates satisfy $|\widehat{\alpha}-\alpha| \le \eta/16$ and $|\widehat{\beta}-\beta| \le \eta/16$. If the completeness hypothesis holds, then $\widehat{\alpha} \ge \eta/2$. If $V' = \{0\}$, the algorithm therefore accepts. Otherwise, $\beta = \alpha|A|/|V'| \ge 2\alpha$, and

$$\widehat{\beta} \ge 2\alpha - \frac{\eta}{16} > \frac{3}{2}\left(\alpha + \frac{\eta}{16}\right) \ge \frac{3}{2}\widehat{\alpha},$$

so the algorithm again accepts with probability at least $1-\delta$. Conversely, suppose that the algorithm accepts on the same concentration event. This implies that $\alpha \ge 7\eta/16$. If $V' = \{0\}$, then $a \in A$, so $\beta = 1$ and $|V'| = 1 \le |A|$. Otherwise, $\beta \ge 11\eta/16$ and

$$\beta - \alpha \ge \frac{1}{2}\widehat{\alpha} - \frac{2\eta}{16} > 0,$$

so $|V'|/|A| = \alpha/\beta < 1$. Therefore, with probability at least $1-\delta$, every accepted candidate satisfies $|V'| \le |A|$ and the required density bounds. □

Proof of Lemma 5.3. Apply Claim 5.4 to obtain an affine coset $a+V'$, and then apply Claim 5.5 to $a+V'$ with $\eta = 2^{-O(R)}$ and failure probability δ, accepting exactly when that test accepts. If $\mathrm{d}[U_A; U_V] \le R$, then Claim 5.4 shows with probability $\Omega(1)$ that $a+V'$ satisfies the completeness hypotheses of Claim 5.5. Conditional on this event, the test accepts with probability at least $1-\delta$, proving completeness. The soundness conclusion follows directly from the soundness part of that claim. The localization step uses $\mathrm{poly}(n)$ time and one sample from U_A, while the test uses $2^{O(R)}\,\mathrm{poly}(n, \log(1/\delta))$ resources. □

6 Proof of main theorem

We prove Theorem 1.2 by combining the GGMT traversal of Section 4 with extraction and verification from Section 5. Each trial constructs algorithmic access to a random variable X_T at the leaf of the GGMT tree, extracts a candidate subspace V, and verifies a coset $a+V'$ with $V' \le V$. Verification lets us amplify the success probability by repeating these steps independently and accepting only cosets with the required

size and overlap bounds. Since several subroutines have a rejection sampling component that might not terminate, we impose a sufficiently large polynomial budget on the time, sample, and membership-query complexity of each trial, aborting and starting a new trial if the budget is exceeded.

Fix sufficiently large effective universal constants C_1, C_2, C_3 and set

$$R := C_1 \log(2K), \qquad N := \left\lceil C_3 (\log(2K))^{C_2} \log \frac{2}{\delta} \right\rceil, \qquad \delta_{\mathrm{ver}} := \frac{\delta}{2N}.$$

Algorithm 4: Algorithmic PFR

Input: Uniform sampling and membership-oracle access to a nonempty set $A \subseteq \mathbb{F}_2^n$ satisfying $|A + A| \le K|A|$, and $0 < \delta < 1$.

Output: An affine coset $a + V'$, represented by a and a basis for V', or failure.

repeat N **times**
- Run Algorithm 2 to obtain algorithmic access to X_T. If it aborts, start the next trial.
- Run Algorithm 3 on X_T with $C = 2d_{\mathrm{term}} + L_0$, under the extraction budgets, to obtain a basis for V. If a budget is exceeded, start the next trial.
- Apply the verification procedure of Lemma 5.3 to V with parameters R and δ_{ver}.
- **if** *verification accepts and outputs* $a + V'$ **then**
  - **return** a and a basis for V'.

return failure.

Proof of Theorem 1.2. We first bound the success probability of one trial from below. By Claim 4.15, with probability $\Omega(1/\mathrm{polylog}(2K))$, the traversal returns algorithmic access to X_T satisfying

$$\mathrm{d}[X_T; X_T] \le 2d_{\mathrm{term}}, \qquad \mathrm{d}[U_A; X_T] = O(\log K), \qquad \Delta[X_T] \le L_0,$$

where d_{term} and L_0 are the absolute constants of Claim 4.12. Conditioning on this event, the hypothesis of Lemma 5.1 holds with $C = 2d_{\mathrm{term}} + L_0$, so extraction returns a basis for $V \le \mathbb{F}_2^n$ satisfying $\mathrm{d}[X_T; U_V] = O(1)$ with probability at least 0.99. Since it makes $\mathrm{poly}(n)$ calls to the algorithmic access of X_T, and each call has $\mathrm{poly}(n, K)$ complexity, for a sufficiently large polynomial budget, Markov's inequality shows that extraction succeeds within the budget with probability at least constant probability. When extraction succeeds, the triangle inequality gives

$$\mathrm{d}[U_A; U_V] \le \mathrm{d}[U_A; X_T] + \mathrm{d}[X_T; U_V] = O(\log(2K)) \le R.$$

By the completeness guarantee of Lemma 5.3, verification then accepts with conditional probability $\Omega(1)$, and therefore one trial accepts with probability $\Omega(1/\mathrm{polylog}(2K))$. By the choice of N, the probability that no trial accepts is at most

$$\left(1 - \frac{1}{C_3(\log(2K))^{C_2}}\right)^N \le \exp\left(-\frac{N}{C_3(\log(2K))^{C_2}}\right) \le \frac{\delta}{2}.$$

The soundness guarantee of Lemma 5.3 applies to every candidate V, including those from unsuccessful traversals or extractions. Except with probability δ_{ver} per verification call, an accepted coset $a + V'$ satisfies

$$|V'| \le |A|, \qquad \frac{|A \cap (a + V')|}{|A|} \ge 2^{-O(R)} = K^{-O(1)}, \qquad \frac{|A \cap (a + V')|}{|V'|} \ge 2^{-O(R)} = K^{-O(1)}.$$

There are at most N verification calls, so the probability that any accepted output violates these bounds is at most $N\delta_{\mathrm{ver}} = \delta/2$. Together with the preceding acceptance bound, this shows that the algorithm returns a coset satisfying all three bounds with probability at least $1 - \delta$.

The imposed budgets for traversal and extraction bound the complexity of each trial by $\mathrm{poly}(n, K)$. Verification uses

$$2^{O(R)}\,\mathrm{poly}\left(n, \log\frac{1}{\delta_{\mathrm{ver}}}\right) = \mathrm{poly}\left(n, K, \log\frac{1}{\delta}\right)$$

time, samples, and membership queries. This proves the desired complexity bounds in Theorem 1.2.

Finally, we prove the covering conclusion via a standard argument. Setting $B := A \cap (a + V')$ and choosing a maximal collection of pairwise disjoint translates $t + B$ with $t \in A$. Since $B \subseteq A$, these translates lie in $A + B \subseteq A + A$, so their number is at most

$$\frac{|A + B|}{|B|} \le \frac{K|A|}{|B|} = K^{O(1)}.$$

For every $x \in A$, maximality ensures that $x + B$ intersects a chosen translate $t + B$. Since $B \subseteq a + V'$, and since we are in characteristic 2, we have that $x \in t + B + B \subseteq t + V'$. The corresponding cosets $t + V'$ therefore cover A, proving the required conclusion for V'. □

7 Quadratic Goldreich-Levin

Theorem 7.1. *Let $\varepsilon, \delta > 0$. There exists a universal constant $c_1, c_2 > 0$ and a randomized algorithm which, given oracle access to a function $f : \mathbb{F}_2^n \to [-1, 1]$, either outputs a quadratic form q or $\perp$ satisfying:*

- ***Completeness:*** *If $\|f\|_{U^3} \ge \varepsilon$, then with probability at least $1 - \delta$, the algorithm outputs a quadratic form q such that*

$$\mathop{\mathbb{E}}_{x}\left[f(x) \cdot (-1)^{q(x)}\right] \ge c_1\varepsilon^{c_2}.$$

- ***Soundness:*** *For every input f, including those not satisfying the promise, the probability of outputting a quadratic form q such that*

$$\mathop{\mathbb{E}}_{x}\left[f(x) \cdot (-1)^{q(x)}\right] < c_1\varepsilon^{c_2}/2$$

  is at most δ.

Further, the algorithm uses $\mathrm{poly}\,(n, 1/\varepsilon, \log 1/\delta)$ *queries and time.*

The remainder of this section is devoted to proving Theorem 7.1. First, Lemma 7.2 uses [TW14, Section 4] to extract from f a function $\varphi : \mathbb{F}_2^n \to \mathbb{F}_2^n$ and a set S of pairs $(x, \varphi(x))$ such that $|\widehat{f_x}(\varphi(x))|$ is large for every $(x, \varphi(x)) \in S$, and S has small doubling constant. This guarantee is saying that $\varphi(x)$ is a large Fourier coefficient of the derivative function f_x. When f is close to quadratic, $\varphi(x)$ is close to linear and therefore the set of pairs $(x, \varphi(x))$ is close to a linear subspace that has high overlap with the set S.

Since S has small doubling constant, we can use the PFR algorithm in Theorem 1.2 to recover a subspace W that covers S with a small number of cosets. From W, we can use Lemma 7.3 to recover a quadratic q that is correlated with f in three steps:

1. First, Claim 7.4 shows that the existence of a large set S with large Fourier coefficients implies the existence of a linear map T such that the average squared derivative Fourier coefficient $\mathbb{E}_x\,|\widehat{f_x}(Tx)|^2$ is large.

2. Then, Claim 7.5 symmetrizes the linear map T to get a symmetric zero-diagonal matrix B such that $\mathbb{E}_x |\widehat{f_x}(Bx)|^2$ is large.

3. Finally, Claim 7.6 converts the symmetric zero-diagonal matrix B into a quadratic polynomial q such that $\mathbb{E}_y \left[f(y)(-1)^{q(y)} \right]$ is large.

The algorithm described so far works for Boolean-valued functions $f : \mathbb{F}_2^n \to \{-1, 1\}$; we use the rounding procedure of [TW14] (see Lemma 7.7) to extend the algorithm to functions $f : \mathbb{F}_2^n \to [-1, 1]$.

Lemma 7.2. *Let $\varepsilon > 0$ be an arbitrary real number. Let $f : \mathbb{F}_2^n \to \{-1, 1\}$ be a function such that $\|f\|_{U^3} \geq \varepsilon$. Then there exists a randomized algorithm that uses $\mathrm{poly}(n, 1/\varepsilon)$ time and queries to f to provide exact membership-oracle access and independent uniform-sampling access to a set $S \subseteq \mathbb{F}_2^n \times \mathbb{F}_2^n$, for which the following properties hold with probability at least $\varepsilon^{O(1)}$:*

- *The set S is large $|S| \geq \varepsilon^{O(1)} \cdot 2^n$ and has small doubling constant, i.e., $|S+S| \leq K|S|$ for some $K = \varepsilon^{-O(1)}$.*
- *There exists a function $\varphi : \mathbb{F}_2^n \to \mathbb{F}_2^n$ such that $S \subseteq \{(x, \varphi(x)) : x \in \mathbb{F}_2^n\}$, and for every $(x, \varphi(x)) \in S$, we have $|\widehat{f_x}(\varphi(x))| \geq \varepsilon^{O(1)}$.*

Further, each call to the constructed membership and uniform-sampling oracles for S uses $\mathrm{poly}(n, 1/\varepsilon)$ expected time and queries to f.

Proof. By Tulsiani and Wolf [TW14, Section 4], given oracle access to f, there exists an algorithm that simulates oracle access to a function $\varphi : \mathbb{F}_2^n \to \mathbb{F}_2^n$, and a randomized algorithm $\mathcal{S}$ such that with probability at least $\varepsilon^{O(1)}$, the following guarantees hold:

- There exist sets $A^- \subseteq A^+ \subseteq \{(x, \varphi(x)) : x \in \mathbb{F}_2^n\}$ such that
$$|A^-| \geq \varepsilon^{O(1)} 2^n, \qquad |A^+ + A^+| \leq \varepsilon^{-O(1)} 2^n,$$
and $|\widehat{f_x}(\varphi(x))| \geq \varepsilon^{O(1)}$ for every $(x, \varphi(x)) \in A^+$.
- For all $(x, \varphi(x)) \in A^-$, the algorithm $\mathcal{S}$ accepts with probability at least $1 - 2^{-n-2}$, and for all $(x, \varphi(x)) \notin A^+$, the algorithm $\mathcal{S}$ accepts with probability at most 2^{-n-2}.

Further, the algorithm that evaluates φ, as well as $\mathcal{S}$ use $\mathrm{poly}(n, 1/\varepsilon)$ time and queries to f. For the rest of this proof, we will condition on the good event that the above guarantees hold.

We fix a random tape for $\mathcal{S}$ so that it is deterministic, and define the set
$$S = \{(x, \varphi(x)) : x \in \mathbb{F}_2^n \text{ and } \mathcal{S}(x, \varphi(x)) = 1\}.$$
A point in A^- is omitted from S with probability at most 2^{-n-2}, while a point outside A^+ is included in S with probability at most 2^{-n-2}. Therefore a union bound gives that conditioned on the good event, with probability at least $1 - 2^n \cdot 2^{-n-2} = \frac{3}{4}$, we have $A^- \subseteq S \subseteq A^+$. On this event, which occurs with probability $\varepsilon^{O(1)}$, we have
$$|S| \geq |A^-| \geq \varepsilon^{O(1)} 2^n,$$
$$|S + S| \leq |A^+ + A^+| \leq \varepsilon^{-O(1)} 2^n \leq \varepsilon^{-O(1)} |S|,$$
and for every $(x, \varphi(x)) \in S$, since $S \subseteq A^+$, we have $|\widehat{f_x}(\varphi(x))| \geq \varepsilon^{O(1)}$.

Finally, we can implement a membership oracle for S as follows: on input (x, y), compute $\varphi(x)$, and check if $y = \varphi(x)$. If not, reject. Otherwise, return the value of $\mathcal{S}(x, \varphi(x))$. A uniform sampler can be implemented by repeatedly drawing a fresh uniform $x \in \mathbb{F}_2^n$ and accepting exactly when $(x, \varphi(x)) \in S$. On the good event above, the expected number of trials until acceptance is at most $\varepsilon^{-O(1)}$. Each membership and sample query therefore uses $\mathrm{poly}(n, 1/\varepsilon)$ time and queries to f. □

Lemma 7.3. *Let $0 < \alpha, \gamma, \delta \le 1$ be arbitrary real numbers, and let $L \ge 1$. Let $f : \mathbb{F}_2^n \to \{-1, 1\}$. Suppose that there is a set $S \subseteq \{(x, \varphi(x)) : x \in \mathbb{F}_2^n\} \subseteq \mathbb{F}_2^n \times \mathbb{F}_2^n$ for some function $\varphi : \mathbb{F}_2^n \to \mathbb{F}_2^n$ such that $|S| \ge \alpha 2^n$, and $|\widehat{f_x}(\varphi(x))| \ge \gamma$ for every $(x, \varphi(x)) \in S$. Suppose further that there exists a linear subspace $W \le \mathbb{F}_2^n \times \mathbb{F}_2^n$, $|W| \le |S|$ such that S is covered by at most L cosets of W.*

Then, there is a randomized algorithm which, given oracle access to f and the basis for W, outputs with probability at least $1 - \delta$, a polynomial $q : \mathbb{F}_2^n \to \mathbb{F}_2$ of degree at most two satisfying

$$\mathop{\mathbb{E}}_{y} \left[f(y)(-1)^{q(y)} \right] \ge \frac{\alpha\gamma^2}{2L^2}.$$

The algorithm uses $\mathrm{poly}(n, L, 1/\alpha, 1/\gamma, \log(1/\delta))$ time and queries to f.

Claim 7.4. *Under the hypotheses of Lemma 7.3, there is an algorithm that computes, from the basis of W, in $\mathrm{poly}(n)$ time, a linear map T such that*

$$\mathop{\mathbb{E}}_{x} \left| \widehat{f_x}(Tx) \right|^2 \ge \frac{\alpha\gamma^2}{L^2}.$$

Proof. Define $\pi(x, y) = x$ to be the first-coordinate projection, and define $\ker \pi_W := \{w \in \mathbb{F}_2^n : (0, w) \in W\}$. The algorithm first computes a basis W of the form

$$(0, w_1), \ldots, (0, w_k), (u_1, v_1), \ldots, (u_\ell, v_\ell),$$

where $\{w_1, \ldots, w_k\}$ is a basis for $\ker \pi_W$, and $\{u_1, \ldots, u_\ell\}$ is a basis for $\pi(W)$. The algorithm then defines the linear map T by setting $T(u_i) = v_i$ for $i \in [\ell]$, and extending T to be zero on the complement of $\pi(W)$ in $\mathbb{F}_2^n$. Therefore, for every $x \in \pi(W)$, we have that $(x, T(x)) \in W$. Such a linear map exists, because the vectors u_i's are linearly independent. This can be done in $\mathrm{poly}(n)$ time using Gaussian elimination.

Since S is covered by at most L cosets of W, there exists a coset $a + W$ such that $|S \cap (a + W)| \ge |S|/L$. Since $(S \cap (a + W)) \subseteq \{(x, \varphi(x)) : x \in \mathbb{F}_2^n\}$, π is injective on $S \cap (a + W)$, and therefore $|S \cap (a + W)| \le |\pi(a + W)| = |\pi(W)| = |W|/|\ker \pi_W|$. Rearranging gives

$$|\ker \pi_W| \le |W|/|S \cap (a + W)| \le |S|/|S \cap (a + W)| \le L. \tag{77}$$

We now claim that linear map T agrees with φ on (the first-coordinate projections of) many points in $S \cap (a + W)$, up to an affine translation. To see this, write $a = (a_1, a_2)$. Then, since we are in characteristic 2, for every $(x, \varphi(x)) \in S \cap (a + W)$, we have that $(x + a_1, \varphi(x) + a_2) \in W$. This means that $x + a_1 \in \pi(W)$, and so $(x + a_1, T(x + a_1)) \in W$. Since both $(x + a_1, \varphi(x) + a_2)$ and $(x + a_1, T(x + a_1))$ are in W, their difference is in $\ker \pi_W$. Therefore, there exists some $w \in \ker \pi_W$ such that

$$\varphi(x) + a_2 = T(x + a_1) + w = Tx + (Ta_1 + w + a_2).$$

By Equation (77), there exists a fixed $w^* \in \ker \pi_W$ for which the above equation holds for at least $|S \cap (a + W)|/|\ker \pi_W| \ge |S|/L^2$ points $(x, \varphi(x)) \in S \cap (a + W)$. Letting $b := Ta_1 + w^* + a_2$, we have concluded that for at least $|S|/L^2$ points $(x, \varphi(x)) \in S$, we have that $\varphi(x) = Tx + b$.

Finally, we have that

$$\mathop{\mathbb{E}}_{x} \left| \widehat{f_x}(Tx) \right|^2 \ge \mathop{\mathbb{E}}_{x} \left| \widehat{f_x}(Tx + b) \right|^2 = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} \left| \widehat{f_x}(Tx + b) \right|^2 \ge \frac{1}{2^n} \cdot \frac{|S|}{L^2} \cdot \gamma^2 \ge \frac{\alpha\gamma^2}{L^2},$$

where first inequality follows by 3.8, finishing the proof. ∎

Claim 7.5 (Lemma 4.16 of [TW14], taken from the proof of Theorem 2.3 of [Sam07])**.** *For a Boolean f and $0 < \eta \le 1$, any linear map T satisfying $\mathbb{E}_x |\widehat{f_x}(Tx)|^2 \ge \eta$ can be converted in* $\mathrm{poly}(n)$ *time into a symmetric zero-diagonal matrix B satisfying*

$$\mathop{\mathbb{E}}_x \left|\widehat{f_x}(Bx)\right|^2 \ge \eta^2.$$

Claim 7.6. *For a Boolean f and $0 < \eta, \delta \le 1$, if a symmetric zero-diagonal matrix B satisfies $\mathbb{E}_x |\widehat{f_x}(Bx)|^2 \ge \eta^2$, then there is a randomized algorithm which, given B and oracle access to f, outputs with probability at least $1 - \delta$ a quadratic q satisfying*

$$\mathop{\mathbb{E}}_y \left[f(y)(-1)^{q(y)}\right] \ge \frac{\eta}{2}.$$

The algorithm uses $\mathrm{poly}(n, 1/\eta, \log(1/\delta))$ *time and queries.*

We use the same integration argument originally described in [GT08] and used in [TW14, Lemma 4.17], followed by a Goldreich–Levin step (see Theorem 3.7) to recover the linear part of the quadratic. We give the algorithm and its analysis here for completeness.

Proof. The algorithm is as follows: given symmetric B with zero diagonal, define the strictly upper triangular matrix M such that $M + M^{\mathsf{T}} = B$. Then, define the function

$$g(y) = f(y)(-1)^{\langle y, My\rangle}$$

and simulate oracle access to g by making one query to f and computing the quadratic phase. Next, run the Goldreich–Levin algorithm on g with threshold $\eta/2$ and failure probability $\delta/2$. This returns a list of $O(\eta^{-2})$ frequencies which, with probability at least $1 - \delta/2$, contains every ξ such that $|\widehat{g}(\xi)| \ge \eta/2$. Finally, using fresh random queries, the algorithm estimates the Fourier coefficient at every frequency in the list to additive error $\eta/8$, with total failure probability at most $\delta/2$, and selects a frequency ξ with largest estimated absolute value. It then chooses $c \in \mathbb{F}_2$ so that multiplying the estimated coefficient by $(-1)^c$ makes it nonnegative, and outputs

$$q(x) := \langle x, Mx\rangle + \langle \xi, x\rangle + c.$$

If the list is empty, the algorithm outputs $\perp$. This completes the description of the algorithm.

We now prove correctness. We first show that g must have a large Fourier coefficient.

$$\max_{\xi \in \mathbb{F}_2^n} |\widehat{g}(\xi)|^2 = \left(\max_{\xi \in \mathbb{F}_2^n} |\widehat{g}(\xi)|^2\right) \sum_{\xi \in \mathbb{F}_2^n} \widehat{g}(\xi)^2 \ge \sum_{\xi \in \mathbb{F}_2^n} \widehat{g}(\xi)^4 = \mathop{\mathbb{E}}_x \left|\mathop{\mathbb{E}}_y [g(y)g(y+x)]\right|^2 \ge \eta^2,$$

where first equality is by Parseval's identity, second equality is by Claim 3.9 and the last inequality follows from the hypothesis, and because

$$\begin{aligned}
\mathop{\mathbb{E}}_y [g(y)g(y+x)] &= \mathop{\mathbb{E}}_y f_x(y)(-1)^{\langle y, My\rangle + \langle y+x, M(y+x)\rangle} \\
&= \mathop{\mathbb{E}}_y f_x(y)(-1)^{\langle x, Mx\rangle + \langle y, (M+M^{\mathsf{T}})x\rangle} \\
&= \mathop{\mathbb{E}}_y f_x(y)(-1)^{\langle x, Mx\rangle + \langle y, Bx\rangle} \\
&= (-1)^{\langle x, Mx\rangle} \widehat{f_x}(Bx).
\end{aligned}$$

Hence some Fourier coefficient of g has magnitude at least η, and when the Goldreich–Levin call and subsequent estimates succeed, the selected frequency ξ has estimated magnitude at least $7\eta/8$, and true magnitude at least $3\eta/4$. The estimated sign is correct, and so $(-1)^c \widehat{g}(\xi) = |\widehat{g}(\xi)| \ge \eta/2$, in particular.

Therefore, the output is a polynomial of degree at most two, and on these success events its correlation with f is

$$\mathop{\mathbb{E}}_{y} f(y)(-1)^{q(y)} = (-1)^{c} \mathop{\mathbb{E}}_{y} g(y)(-1)^{\langle \xi, y\rangle} = (-1)^{c}\widehat{g}(\xi) = |\widehat{g}(\xi)| \geq \frac{\eta}{2}.$$

The Goldreich–Levin call and the simultaneous estimates each fail with probability at most $\delta/2$. A union bound therefore gives overall success probability at least $1-\delta$. Constructing M takes $\mathrm{poly}(n)$ time. Goldreich–Levin uses $\mathrm{poly}(n, 1/\eta, \log(1/\delta))$ time and queries, and Chernoff's bound followed by a union bound over the $O(\eta^{-2})$ outputted frequencies shows that the subsequent Fourier estimation can be done within the same polynomial resource bound. □

Proof of Lemma 7.3. Setting $\eta := \frac{\alpha\gamma^2}{L^2}$, Claim 7.4 computes a linear map T with $\mathbb{E}_x \left|\widehat{f_x}(Tx)\right|^2 \geq \eta$. Applying Claim 7.5 to T gives a symmetric zero-diagonal matrix B with $\mathbb{E}_x \left|\widehat{f_x}(Bx)\right|^2 \geq \eta^2$. Finally, Claim 7.6 returns, with probability at least $1-\delta$, a quadratic q whose correlation with f is at least $\eta/2$. All these algorithms use $\mathrm{poly}(n, 1/\eta, \log(1/\delta))$ time and queries. □

Lemma 7.7 (Boolean rounding)**.** *Let $0 < \varepsilon, \eta, \delta \leq 1$. Suppose there is a randomized algorithm which, given oracle access to any function $\widetilde{f} : \mathbb{F}_2^n \to \{-1,1\}$ satisfying $\|\widetilde{f}\|_{U^3} \geq \varepsilon/2$, outputs, with probability at least $1-\delta$, a quadratic polynomial q such that*

$$\mathop{\mathbb{E}}_{x}[\widetilde{f}(x)(-1)^{q(x)}] \geq \eta.$$

Then there is a randomized algorithm which, given oracle access to any function $f : \mathbb{F}_2^n \to [-1,1]$ satisfying $\|f\|_{U^3} \geq \varepsilon$, outputs, with probability at least $1-2\delta$, a quadratic polynomial q such that

$$\mathop{\mathbb{E}}_{x}[f(x)(-1)^{q(x)}] \geq \eta/2.$$

The new algorithm has the same running time and query complexity as the Boolean algorithm, up to polynomial overhead in n, $1/\varepsilon$, $1/\eta$, and $\log(1/\delta)$.

Proof. Independently for each $x \in \mathbb{F}_2^n$, define a random Boolean function $\widetilde{f}$ by

$$\Pr\left[\widetilde{f}(x) = 1\right] = \frac{1+f(x)}{2}.$$

The algorithm implements oracle access to $\widetilde{f}$ lazily: the first time x is queried, it samples $\widetilde{f}(x)$ using one query to $f(x)$ and caches the result. Thus repeated queries receive the same answer and the oracle has exactly the law of one fixed random Boolean function. We use the concentration argument from [TW14, Lemma 3.3]. Since $\mathbb{E}[\widetilde{f}(x)] = f(x)$ and there are only $2^{O(n^2)} = \exp(o(2^n))$ quadratic phases, with probability at least $1-\delta$ the random function $\widetilde{f}$ simultaneously satisfies

$$\|\widetilde{f}\|_{U^3} \geq \|f\|_{U^3} - \varepsilon/2$$

and, for every quadratic polynomial q,

$$\left|\mathop{\mathbb{E}}_{x}\big(\widetilde{f}(x) - f(x)\big)(-1)^{q(x)}\right| \leq \eta/2.$$

The first inequality follows by applying concentration to the degree-eight polynomial defining the eighth power of the U^3 norm. The second follows from Hoeffding's inequality for each fixed quadratic phase, followed by a union bound over all quadratic phases.

Assume that this concentration event occurs. Since $\|f\|_{U^3} \geq \varepsilon$, we have $\|\widetilde{f}\|_{U^3} \geq \varepsilon/2$, so the Boolean algorithm returns, with probability at least $1-\delta$, a quadratic polynomial q satisfying

$$\mathop{\mathbb{E}}_x[\widetilde{f}(x)(-1)^{q(x)}] \geq \eta.$$

For the same q, the uniform correlation bound above gives

$$\begin{aligned}\mathop{\mathbb{E}}_x[f(x)(-1)^{q(x)}] &\geq \mathop{\mathbb{E}}_x[\widetilde{f}(x)(-1)^{q(x)}] - \left|\mathop{\mathbb{E}}_x\big(\widetilde{f}(x)-f(x)\big)(-1)^{q(x)}\right| \\ &\geq \eta - \frac{\eta}{2} = \frac{\eta}{2}.\end{aligned}$$

A union bound over the concentration event and the Boolean algorithm gives success probability at least $1-2\delta$. Finally, each new query to $\widetilde{f}$ uses one query to f, and cached values can be retrieved in polynomial time. The concentration argument is used only in the analysis and requires no additional oracle queries, which proves the claimed complexity bound. □

Proof of Theorem 7.1. First, we handle the case where $f : \mathbb{F}_2^n \to \{-1,1\}$ is Boolean-valued. By Lemma 7.2, with probability $\varepsilon^{O(1)}$ we construct oracle access to a set $S \subseteq \mathbb{F}_2^{2n}$ such that

$$|S| \geq \alpha 2^n, \qquad |S+S| \leq K|S|, \qquad \alpha, \gamma \geq \varepsilon^{O(1)}, \qquad K \leq \varepsilon^{-O(1)},$$

and every $(x, \varphi(x)) \in S$ satisfies $|\widehat{f_x}(\varphi(x))| \geq \gamma$. Conditioning on the event above, we apply Theorem 1.2 to S, with any fixed constant failure probability. It returns a subspace W with $|W| \leq |S|$ such that S is covered by at most $L = K^{O(1)} = \varepsilon^{-O(1)}$ cosets of W. Finally, applying Lemma 7.3 with a fixed constant failure probability, we obtain a quadratic polynomial q satisfying

$$\mathop{\mathbb{E}}_x f(x)(-1)^{q(x)} \geq \alpha\gamma^2/2L^2 \geq \varepsilon^{O(1)}.$$

Thus one trial finds a quadratically correlated phase with probability $\varepsilon^{O(1)}$ in expected cost $\mathrm{poly}(n, 1/\varepsilon)$. (The expected cost comes from sampling from S.) Truncating each trial to a sufficiently large polynomial number of steps, and repeating $\varepsilon^{-O(1)}\log(1/\delta)$ independent trials, we obtain a quadratic with correlation at least $\varepsilon^{O(1)}$ with probability at least $1-\delta/2$. Using fresh uniform queries, we can estimate the correlation of each candidate quadratic with f, obtaining the stated completeness and soundness guarantees for the Boolean-valued case. Finally, we bootstrap the Boolean-valued algorithm to functions $f : \mathbb{F}_2^n \to [-1,1]$ by Lemma 7.7. Estimating the correlation of the returned quadratic with f to additive error at most half of the Boolean correlation guarantee gives the stated completeness and soundness guarantees for the general case. The total time and query complexity are $\mathrm{poly}(n, 1/\varepsilon, \log(1/\delta))$. □

8 Improper agnostic tomography of stabilizer states

In this section, we will present some applications of algorithmic PFR theorem to quantum learning. It was established in [AD26a, AD26b] that *approximate* algorithmic PFR in polynomial time has implications for learning stabilizer-like states. We first state the main combinatorial approximate algorithmic PFR theorem and prove it, before stating our applications. To state our theorem, we need the notion of approximately uniform distributions.

Definition 8.1 (Approximately uniform distribution)**.** *Let A be finite and nonempty. A distribution μ supported on A is R-uniform if for every $x \in A$, we have*

$$\frac{1}{R|A|} \leq \mu(x) \leq \frac{R}{|A|}$$

In approximate algorithmic PFR theorem, we do not have access to uniform samples from A, but rather have access to samples from R-uniform distributions supported on A (observe that $R = 1$ is precisely uniform sampling access to A). We show that as long as R is polynomial in the other parameters, one can recover the same complexity as the usual algorithmic PFR theorem that we proved above.

Theorem 8.2 (Approximate algorithmic PFR)**.** *Let $0 < \delta < 1$, and let $A \subseteq \mathbb{F}_2^n$ be nonempty and satisfy $|A+A| \le K|A|$. Given membership-oracle access to A and independent samples from an R-uniform distribution on A, there is a randomized algorithm that, with probability at least $1-\delta$, outputs a basis for a subspace $V \le \mathbb{F}_2^n$ such that $|V| \le |A|$ and A can be covered by at most $K^{O(1)}$ translates of V. The algorithm uses* $\mathrm{poly}(n, K, R, \log(1/\delta))$ *time, samples, and membership queries.*

8.1 Algorithmic PFR with approximate sampling access

We first record that, when A has small doubling, samples from an R-uniform distribution can be efficiently converted into nearly uniform samples from A.

Lemma 8.3. *Let $A \subseteq \mathbb{F}_2^m$ be nonempty with $|A+A| \le K|A|$, and let μ be an R-uniform distribution on A. Given membership access to A and independent samples from μ, for every $\zeta \in (0, 1/2)$ one can sample from a distribution $\widetilde{U}_A$ satisfying*

$$\|\widetilde{U}_A - U_A\|_{\mathrm{TV}} \le \zeta$$

using $\widetilde{O}\big(R^6K^2/\zeta^2\big)$ samples from μ and membership queries.

Proof. Consider the following lazy random walk on A. Starting from $x \in A$, with probability $1/2$ remain at x; otherwise sample $U, V \sim \mu$, set $y = x + U + V$, and move to y if $y \in A$. Write

$$\nu(g) := \Pr_{U,V\sim\mu}[U+V=g].$$

For distinct $x, y \in A$, the transition probability is

$$P(x,y) = \frac{1}{2}\nu(x+y) = P(y,x),$$

so the uniform distribution U_A is stationary and the chain is reversible. Moreover, since

$$\mu(x) \le \frac{R}{|A|} = R\,U_A(x),$$

the initial distribution μ is an R-warm start with respect to U_A by definition. For $B, C \subseteq A$, define the stationary flow from B to C by

$$Q(B,C) := \sum_{x\in B}\sum_{y\in C} U_A(x)P(x,y).$$

Thus $Q(B, A\setminus B)/U_A(B)$ is the probability that one step leaves B when the current state is uniform on B. A small value means that B can trap the walk for a long time. To allow us to ignore very small sets, for $s \in (0, 1/2)$ define the *s-conductance* as

$$\Phi_s := \min_{\substack{B\subseteq A\\ s<U_A(B)\le 1/2}} \frac{Q(B, A\setminus B)}{U_A(B)-s}.$$

Below we use a standard warm-start mixing bound shows that a lower bound on Φ_s, together with the R-warmness above, gives an explicit bound on the number of steps needed for the walk to approach U_A in

total variation distance. It therefore remains to lower-bound Φ_s. For $B \subseteq A$, write $C = A \setminus B$, $N = |A|$, and $b = |B|/N$. Let

$$r_{B,C}(g) := |\{(x, y) \in B \times C : x + y = g\}|.$$

Since μ is R-uniform, every $u \in A$ satisfies $\mu(u) \geq \frac{1}{RN}$. Hence, for any $g \in \mathbb{F}_2^m$,

$$\nu(g) = \Pr_{U,V\sim\mu}[U + V = g] = \sum_{\substack{u,v\in A \\ u+v=g}} \mu(u)\mu(v) \geq \sum_{\substack{u,v\in A \\ u+v=g}} \frac{1}{R^2N^2} = \frac{r_A(g)}{R^2N^2}.$$

Moreover, every representation $g = x + y$ with $(x, y) \in B \times C$ gives the distinct reversed representation $g = y + x$ with $(y, x) \in C \times B$. Since B and C are disjoint, these two ordered representations are distinct. Therefore

$$r_A(g) \geq r_{B,C}(g) + r_{C,B}(g) = 2r_{B,C}(g),$$

and hence

$$\nu(g) \geq \frac{2r_{B,C}(g)}{R^2N^2}.$$

Hence the stationary flow across the cut satisfies

$$Q(B, C) = \frac{1}{2N} \sum_g r_{B,C}(g)\nu(g) \geq \frac{1}{R^2N^3} \sum_g r_{B,C}(g)^2. \tag{78}$$

The support of $r_{B,C}$ is contained in $B + C \subseteq A + A$, and therefore Cauchy–Schwarz gives

$$\sum_g r_{B,C}(g)^2 \geq \frac{|B|^2|C|^2}{|B + C|} \geq \frac{|B|^2|C|^2}{KN}.$$

Plugging into Eq. (78) gives

$$Q(B, C) \geq \frac{b^2(1 - b)^2}{R^2K}.$$

Recall that we proved

$$Q(B, A \setminus B) \geq \frac{b^2(1 - b)^2}{R^2K}, \qquad b := \frac{|B|}{N}.$$

Now fix a set B appearing in the definition of Φ_s, so that $s < b \leq \frac{1}{2}$. Since $b \leq 1/2$, we have $1 - b \geq 1/2$, and therefore $(1 - b)^2 \geq \frac{1}{4}$. Hence

$$\frac{Q(B, A \setminus B)}{b - s} \geq \frac{b^2(1 - b)^2}{R^2K(b - s)} \geq \frac{b^2}{4R^2K(b - s)} \geq \frac{s}{R^2K}$$

where the final inequality used $b^2 - 4s(b - s) = (b - 2s)^2 \geq 0$. Therefore every set B in the minimization defining Φ_s satisfies

$$\frac{Q(B, A \setminus B)}{b - s} \geq \frac{s}{R^2K}.$$

Taking the minimum over all such B gives

$$\Phi_s \geq \frac{s}{R^2K}.$$

We will use the following standard warm-start form of the Lovász–Simonovits mixing bound [LS93].

Lemma 8.4 (Warm-start mixing bound)**.** *Let P be a finite lazy reversible Markov chain with stationary distribution π. Suppose the initial distribution σ is M-warm with respect to π, meaning for every state x, $\sigma(x) \le M\pi(x)$. For $s \in (0, 1/2)$, let*

$$\Phi_s := \min_{B : s < \pi(B) \le 1/2} \frac{Q(B, B^c)}{\pi(B) - s},$$

where $Q(B, C) := \sum_{x \in B} \sum_{y \in C} \pi(x) P(x, y)$. Then, for every $t \ge 0$,

$$\|\sigma P^t - \pi\|_{\mathrm{TV}} \le Ms + M \exp\left(-\frac{\Phi_s^2 t}{2}\right).$$

As we saw above, μ is R-warm with respect to U_A, we may apply Lemma 8.4. Taking $s = \zeta/(2R)$ and using

$$\Phi_s \ge \frac{s}{R^2 K} = \frac{\zeta}{2R^3 K},$$

we obtain

$$\|\mu P^t - U_A\|_{\mathrm{TV}} \le \frac{\zeta}{2} + R \exp\left(-\frac{\zeta^2 t}{8R^6 K^2}\right).$$

Hence the total-variation distance is at most ζ once

$$t \ge \frac{8R^6 K^2}{\zeta^2} \log \frac{2R}{\zeta}.$$

Thus $t = poly(R, K, 1/\zeta, \log(R/\zeta))$ steps suffice. Each step uses two fresh samples from μ and one query. □

We can now transfer any polynomial-time PFR algorithm requiring uniform samples to the approximately uniform-sample setting.

Proof of Theorem 8.2. Run our main algorithmic PFR results algorithm, replacing each requested uniform sample from A by an independent output of Lemma 8.3. Let

$$Q := (mK + 2)^{C_{\mathrm{PFR}}}$$

be an upper bound on the total number of samples requested by the PFR algorithm, and generate each replacement sample to total-variation accuracy $\zeta := 1/100Q$. By the coupling characterization of total variation distance, each replacement sample can be coupled to an exactly uniform sample so that they disagree with probability at most ζ. A union bound over the at most Q sample requests therefore couples the entire execution to the exact-uniform execution with failure probability at most $Q\zeta \le 1/100$. Consequently the success probability is at least $2/3 - 1/100 \ge 3/5$. By Lemma 8.3, each replacement sample uses

$$O\big(R^6 K^2 Q^2 \log(RQ)\big)$$

operations and oracle calls. Since Q itself is polynomial in n and K, the total complexity is bounded by $\mathrm{poly}(n, K, R)$ The structural guarantee on H is exactly that supplied by our algorithmic PFR theorem. □

8.2 Quantum implications

We present some implication of the algorithmic PFR theorem in this section.

8.2.1 Self-correction.

In a recent work, [AD26a] considered the task of *self-correction*, a weakening of agnostic learning. Recall that in agnostic learning, for an unknown state $|\psi\rangle$ which is promised to be opt-close to a state in a class C, the goal is, given copies of $|\psi\rangle$ output an element of C which is opt $-\varepsilon$ close to $|\psi\rangle$. In the self-correction task, they consider if the task becomes easier if the goal was to output an element in C which is poly(opt) close to $|\psi\rangle$. To this end, they proved the following theorem.[14]

Theorem 8.5 ([AD26a]). *Let $\tau > 0$. Let $|\psi\rangle$ be an unknown n-qubit quantum state such that*

$$\max_{|\varphi\rangle \in \text{stabilizer}} |\langle\psi|\varphi\rangle|^2 \geq \tau.$$

Assuming the approximate algorithmic PFR conjecture, there is a protocol that with probability $1-\delta$, outputs a $|\varphi\rangle \in$ stabilizer such that $|\langle\varphi|\psi\rangle|^2 \geq \tau^C$ (for a universal constant $C > 1$) using $\mathrm{poly}(n, 1/\tau, \log(1/\delta))$ time and copies of $|\psi\rangle$.

Using our approximate algorithm PFR result in Theorem 8.2, we remove the assumption in their work

Corollary 8.6. *The guarantees of Theorem 8.5 hold without any assumptions.*

8.2.2 Improper agnostic tomography.

In a later work, [AD26b] provided a boosting framework that took the self-correction procedure to give an algorithm for learning stabilizer decompositions of any quantum state $|\psi\rangle$ given copies. Particularly, the following is now true as a consequence of Corollary 8.6 applied to [AD26b, Theorem 1.2].

Corollary 8.7. *Let $\varepsilon, \delta \in (0,1)$. Suppose $|\psi\rangle$ is an unknown n-qubit state. Then, there is an algorithm that uses copies of $|\psi\rangle$ and with probability $\geq 1-\delta$, outputs a list of $\kappa = \mathrm{poly}(1/\varepsilon)$ many states $\{|\varphi_i\rangle\}_{i\in[\kappa]}$ and coefficients $\beta \in \mathbb{C}^k$ such that $|\psi\rangle$ can be expressed (up to a global phase) as*

$$|\psi\rangle = \sum_{i=1}^{\kappa} \beta_i |\varphi_i\rangle + \alpha |\varphi_R\rangle, \text{ where } |\alpha|^2 \cdot \max_{|\varphi\rangle \in \text{stabilizer}} |\langle\varphi|\psi\rangle|^2 \leq \varepsilon.$$

The sample and time complexity of this algorithm is $\mathrm{poly}(n, 1/\varepsilon)$.

Moreover, this implies a polynomial-time agnostic learner for the class of stabilizer states, albeit the output of the agnostic learner is *improper*, i.e., need not be a stabilizer state itself. The following was shown in [AD26b].

Theorem 8.8 ([AD26b, Theorem 1.3]). *Let $\delta, \varepsilon \leq \mathrm{opt} \in (0,1)$. Suppose $|\psi\rangle$ is an unknown n-qubit state with (unknown) optimal stabilizer fidelity*

$$\mathrm{opt} = \max_{|\varphi\rangle \in \text{stabilizer}} |\langle\psi|\varphi\rangle|^2$$

Assuming that Self-correction is solvable in $\mathrm{poly}(n, 1/\varepsilon)$ time, there is an algorithm, that with probability $\geq 1-\delta$, uses copies of $|\psi\rangle$ to output a state $|\varphi\rangle$, which is a sum of $\kappa = \mathrm{poly}(1/\varepsilon)$ many $|\varphi_i\rangle \in$ stabilizer such that

$$|\langle\varphi|\psi\rangle|^2 \geq \mathrm{opt} - \varepsilon,$$

[14] We remark that in [AD26a], they invoke the algorithmic PFR under the sampling access arising from its BSG reduction reduction step, while our theorem is stated for approximately-uniform sampling distributions. This causes no essential difference since one can use the quantitative bounds in the BSG construction in their paper to imply that, after conditioning on the relevant set, the induced sampling distribution is $\gamma^{-O(1)}$-flat. Thus our approximate PFR algorithm applies with only $\mathrm{poly}(1/\gamma)$ overhead, and the remainder of the self-correction argument in [AD26a] goes through unchanged.

The sample and time complexity of the algorithm is $\mathrm{poly}(n, 1/\varepsilon, \log 1/\delta)$. [15]

Since Theorem 8.2 gave a polynomial-time algorithm for self-correction, we also obtain an improper agnostic learner for stabilizer states.

Corollary 8.9. *The guarantees of Theorem 8.8 hold without any assumptions.*

8.2.3 Learning states with bounded extent

Finally, in [AD26b] they also used the boosting framework on top of self-correction to obtain a tomography algorithm for the class of states with stabilizer extent ξ. For an arbitrary state $|\psi\rangle$, we define the stabilizer extent of $|\psi\rangle$as[16]

$$\xi(|\psi\rangle) = \min\left\{ \sum_i |c_i| : |\psi\rangle = \sum_i c_i |\varphi_i\rangle, |\varphi_i\rangle \in \text{stabilizer} \right\}. \tag{79}$$

Theorem 8.10 ([AD26b])**.** *Let* $\xi > 0, \varepsilon \in (0,1)$*. Suppose* $|\psi\rangle$ *is an unknown n-qubit state with stabilizer extent* $\xi(|\psi\rangle) \leq \xi$*. Then, there exists an algorithm that learns* $|\psi\rangle$ *up to trace distance* ε *in time and sample complexity* $\mathrm{poly}(n, \xi, 1/\varepsilon)$ assuming *an algorithmic PFR conjecture. Furthermore, every* 0.1*-error tomography protocol needs* $\Omega(n + \xi^2)$ *copies.*

Using our approximate algorithm PFR result in Theorem 8.2, we remove the assumption in their work

Corollary 8.11. *The guarantees of Theorem 8.10 hold without any assumptions.*

8.2.4 High stabilizer-dimension states

We have so far described tomography results for stabilizer states or quantum states that admit stabilizer structure. We now turn our attention to states with high stabilizer dimension. We say that an n-qubit pure quantum state $|\psi\rangle$ has stabilizer dimension of k if $|\psi\rangle$ is stabilized by an Abelian group of 2^k Pauli operators and denote this class of states as $\mathcal{S}(k)$. We then have the following self-correction protocol for states with stabilizer dimension $n - t$ i.e., $\mathcal{S}(n-t)$.

Corollary 8.12. *Let* $\varepsilon, \delta \in (0,1)$ *and* $0 \leq t \leq n$*. Suppose that* $|\psi\rangle$ *is an unknown n-qubit state with the promise* $\max_{|\varphi\rangle \in \mathcal{S}(n-t)} |\langle\varphi|\psi\rangle|^2 \geq \tau$*. Then, there is an algorithm that uses copies of* $|\psi\rangle$ *to output a stabilizer state* $|\varphi\rangle$ *such that with probability* $\geq 1 - \delta$*,*

$$|\langle\varphi|\psi\rangle|^2 \geq \mathrm{poly}(2^{-t}\tau).$$

The algorithm consumes sample and time complexity $\mathrm{poly}(n, 2^t, 1/\tau)$.

We quickly remark that the promise of $\mathrm{poly}(2^{-t}\tau)$ is a consequence of the local inverse theorem of Gowers-3 norm of quantum states and refer the reader to [AD26a, Section 5].

Using the boosting framework of [AD26b], we can then use the above self-correction result to obtain an improper agnostic learner of states with high stabilizer dimension.

[15] We remark that their main result is stated in terms of *model class*, which has three requirements: (i) the class is efficiently representable, (ii) it admits a self-correction algorithm, (iii) there is an efficient circuit that can prepare the states in the class. For the class of stabilizer state, (i, iii) are well-known in literature and only (ii) was a bottleneck, which is stated explicitly in this theorem statement.

[16] We remark that in literature [BSS16, BBC+19] extent was defined the *squared* ℓ_1 norm. For the purposes of this paper, the squared ℓ_1 norm or the ℓ_1 norm as defined below will not change the main results.

Corollary 8.13. *Let $\delta, \varepsilon \leq \mathrm{opt} \in (0,1)$ and $0 \leq t \leq n$. Suppose $|\psi\rangle$ is an unknown n-qubit state with (unknown) optimal fidelity*

$$\mathrm{opt} = \max_{|\varphi\rangle \in \mathcal{S}(n-t)} |\langle\psi|\varphi\rangle|^2.$$

Then, there is an algorithm that with probability $\geq 1-\delta$, uses copies of $|\psi\rangle$ to output a state $|\varphi\rangle$, which is a sum of $\kappa = \mathrm{poly}(2^t/\varepsilon)$ many $|\varphi_i\rangle \in$ stabilizer such that

$$|\langle\varphi|\psi\rangle|^2 \geq \mathrm{opt} - \varepsilon,$$

The sample and time complexity of the algorithm is $\mathrm{poly}(n, 2^t, 1/\varepsilon, \log 1/\delta)$.

Note that the previously best known result for agnostic learning of states in $\mathcal{S}(n-t)$ had time complexity $\mathrm{poly}(n(2^t/\varepsilon)^{O(\log(1/\varepsilon))})$ [CGYZ25] and we are able to remove this quasipolynomial dependence on ε. Moreover, Corollary 8.12 can be utilized to learn states which have bounded extent ξ with respect to $\mathcal{S}(n-t)$ in $\mathrm{poly}(n \cdot \xi/\varepsilon)$ time.

8.2.5 Learning Clifford structure in unitaries and Hamiltonians

We now describe some implications of the polynomial-time self-correction protocol of Corollary 8.6 for agnostic tomography of Clifford unitaries and tomography of Clifford-structured objects. Note that Clifford unitaries have previously been shown to be efficiently testable [GNW21, HBvD+26]. We briefly introduce some relevant notation. We will utilize the Choi–Jamiolkowski isomorphism and denote the Choi state of an n-qubit unitary U as $|U\rangle\!\rangle := (U \otimes I)\,|\mathrm{EPR}_n\rangle$ where $|\mathrm{EPR}_n\rangle = 2^{-n/2}\sum_{x\in\mathbb{F}_2^n} |x\rangle\,|x\rangle$ is the state over n many EPR pairs. We will say that U has Clifford fidelity opt if $\max_{V\in\mathrm{Clifford}} |\langle\!\langle V|U\rangle\!\rangle|^2 = \mathrm{opt}$.

As a consequence of Corollary 8.6 and Theorem A.1 of [DJJ+26], the following is now true regarding polynomial-time self-correction of n-qubit Clifford unitaries.

Corollary 8.14. *Let $\tau, \delta \in (0,1)$. Given an unknown n-qubit unitary U that has Clifford fidelity $\geq \tau$, there is a quantum algorithm that with probability $\geq 1-\delta$ outputs a Clifford unitary V such that $|\langle\!\langle V|U\rangle\!\rangle|^2 \geq \mathrm{poly}(\tau)$. The algorithm uses* $\mathrm{poly}(n, 1/\tau)$ *query and time complexity.*

We can then use the above result to give polynomial-time algorithms for learning Clifford decompositions of unitaries and learning unitaries with bounded Clifford extent. We will not give the details here and refer the reader to [DJJ+26, Section 4]. We however quickly state the following implication regarding tomography of n-qubit Hamiltonians which are Clifford-structured, obtained from applying Corollary 8.14 to [DJJ+26, Theorem 4.4].

Corollary 8.15. *Let $\varepsilon, \delta \in (0,1)$ and $\xi \geq 1$. Suppose H is an unknown n-qubit Hamiltonian wwith the promise $\xi = \min\{\|c\|_1 : H = \sum_i c_i V_i$ where $V_i \in$ Cliffords$\}$ and satisfies $\mathrm{Tr}[H] = 0$. Then, there is an algorithm that given query access to $\exp(-iHt)$ outputs $\widehat{H}$ expressable as a linear combination of $\mathrm{poly}(\xi/\varepsilon)$ many Clifford unitaries with probability $\geq 1-\delta$ such that*

$$\|\widehat{H} - H\|_{\overline{2}} \leq \varepsilon,$$

where $\|A\|_{\overline{2}} = \sqrt{2^{-n}\,\mathrm{Tr}\left[A^\dagger A\right]}$ is normalized Frobenius norm. The algorithm uses $t = \mathrm{poly}(\varepsilon/\xi)$ and total time evolution of $\mathrm{poly}(n \cdot \xi/\varepsilon \log(1/\delta))$. The algorithm uses query and time complexity $\mathrm{poly}(n \cdot \xi/\varepsilon \log(1/\delta))$.

References

[ACDG26] Srinivasan Arunachalam, Davi Castro-Silva, Arkopal Dutt, and Tom Gur. Classical and Quantum Polynomial Freiman-Ruzsa Algorithms. In *17th Innovations in Theoretical Computer Science Conference (ITCS 2026)*, pages 11–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2026. [p. 4]

[AD25] Srinivasan Arunachalam and Arkopal Dutt. Polynomial-time tolerant testing stabilizer states. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, page 1234–1241. Association for Computing Machinery, 2025. [pp. 3, 4, 6]

[AD26a] Srinivasan Arunachalam and Arkopal Dutt. Learning stabilizer structure of quantum states. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing*, STOC '26, page 1949–1959, New York, NY, USA, 2026. Association for Computing Machinery. [pp. 3, 4, 6, 7, 53, 57, 58]

[AD26b] Srinivasan Arunachalam and Arkopal Dutt. Tomography of quantum states with bounded extent. *arXiv preprint arXiv:2606.07425*, 2026. [pp. 3, 6, 7, 53, 57, 58]

[ADL14] Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, page 774–783. Association for Computing Machinery, 2014. [p. 3]

[AGG+24] Vahid R Asadi, Alexander Golovnev, Tom Gur, Igor Shinkar, and Sathyawageeswar Subramanian. Quantum worst-case to average-case reductions for all linear problems. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2535–2567. SIAM, 2024. [p. 3]

[AGGS22] Vahid R. Asadi, Alexander Golovnev, Tom Gur, and Igor Shinkar. Worst-case to average-case reductions via additive combinatorics. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 1566–1574. Association for Computing Machinery, 2022. [p. 3]

[BBC+19] Sergey Bravyi, Dan Browne, Padraic Calpin, Earl Campbell, David Gosset, and Mark Howard. Simulation of quantum circuits by low-rank stabilizer decompositions. *Quantum*, 3:181, 2019. [p. 58]

[BC26] Jop Briët and Davi Castro-Silva. A near-optimal quadratic Goldreich-Levin algorithm. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 6233–6239. SIAM, 2026. [pp. 4, 5]

[BDL13] Abhishek Bhowmick, Zeev Dvir, and Shachar Lovett. New bounds for matching vector families. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, page 823–832. Association for Computing Machinery, 2013. [p. 3]

[BFF+01] Tugkan Batu, Eldar Fischer, Lance Fortnow, Ravi Kumar, Ronitt Rubinfeld, and Patrick White. Testing random variables for independence and identity. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 442–451. IEEE, 2001. [p. 12]

[BLR14] Eli Ben-Sasson, Shachar Lovett, and Noga Ron-Zewi. An additive combinatorics approach relating rank to communication complexity. *Journal of the ACM (JACM)*, 61(4):1–18, 2014. [p. 3]

[BNOZ25] Benjamin Bedert, Tamio-Vesa Nakajima, Karolina Okrasa, and Stanislav Zivný. Strong sparsification for 1-in-3-sat via Polynomial Freiman-Ruzsa. In *2025 IEEE 66th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2470–2479, 2025. [p. 3]

[BSS16] Sergey Bravyi, Graeme Smith, and John A Smolin. Trading classical and quantum computational resources. *Physical Review X*, 6(2):021043, 2016. [p. 58]

[BvDH25] Zongbo Bao, Philippe van Dordrecht, and Jonas Helsen. Tolerant testing of stabilizer states with a polynomial gap via a generalized uncertainty relation. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, page 1254–1262. Association for Computing Machinery, 2025. [p. 3]

[CBA+26] Davi Castro-Silva, Jop Briët, Srinivasan Arunachalam, Arkopal Dutt, and Tom Gur. An algorithmic Polynomial Freiman-Ruzsa theorem. *arXiv preprint arXiv:2604.04547*, 2026. [p. 4]

[CFMdW10] Sourav Chakraborty, Eldar Fischer, Arie Matsliah, and Ronald de Wolf. New results on quantum property testing, 2010. [p. 12]

[CGYZ25] Sitan Chen, Weiyuan Gong, Qi Ye, and Zhihan Zhang. Stabilizer bootstrapping: A recipe for efficient agnostic tomography and magic estimation. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, page 429–438, New York, NY, USA, 2025. Association for Computing Machinery. [pp. 4, 6, 7, 59]

[DJJ+26] Arkopal Dutt, Dale Jacobs, John Jeang, Saeed Mehraban, and Vladimir Podolskii. Learning clifford-structured quantum unitaries and hamiltonians. *arXiv preprint arXiv:2608.09912*, 2026. [pp. 7, 59]

[Fel09] Vitaly Feldman. Distribution-specific agnostic boosting. *arXiv preprint arXiv:0909.2927*, 2009. [p. 5]

[GGMT24] W Timothy Gowers, Ben Green, Freddie Manners, and Terence Tao. Marton's Conjecture in abelian groups with bounded torsion. *arXiv preprint arXiv:2404.02244*, 2024. [pp. 3, 7, 10, 20]

[GGMT25] William Timothy Gowers, Ben Green, Freddie Manners, and Terence Tao. On a conjecture of marton. *Annals of Mathematics*, 201(2):515–549, 2025. [pp. 3, 7, 8, 9, 10, 15, 16]

[GIKL24] Sabee Grewal, Vishnu Iyer, William Kretschmer, and Daniel Liang. Improved stabilizer estimation via Bell difference sampling. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, page 1352–1363, New York, NY, USA, 2024. Association for Computing Machinery. [p. 6]

[GIKL25] Sabee Grewal, Vishnu Iyer, William Kretschmer, and Daniel Liang. Efficient Learning of Quantum States Prepared With Few Non-Clifford Gates. *Quantum*, 9:1907, November 2025. [pp. 6, 7]

[GIKL26] Sabee Grewal, Vishnu Iyer, William Kretschmer, and Daniel Liang. Agnostic Tomography of Stabilizer Product States. *Quantum*, 10:2027, March 2026. [p. 6]

[GL89] Oded Goldreich and Leonid A Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32, 1989. [pp. 4, 17]

[GNW21] David Gross, Sepehr Nezami, and Michael Walter. Schur–Weyl duality for the Clifford group with applications: Property testing, a robust Hudson theorem, and de Finetti representations. *Communications in Mathematical Physics*, 385(3):1325–1393, 2021. [pp. 6, 59]

[GT08] Ben Green and Terence Tao. An inverse theorem for the gowers norm. *Proceedings of the Edinburgh Mathematical Society*, 51(1):73–153, 2008. [p. 51]

[GT09] Ben Green and Terence Tao. Freiman's theorem in finite fields via extremal set theory. *Combinatorics, Probability and Computing*, 18(3):335–355, 2009. [p. 3]

[HBvD+26] Marcel Hinsche, Zongbo Bao, Philippe van Dordrecht, Jens Eisert, Jop Briët, and Jonas Helsen. Clifford testing: Algorithms and lower bounds. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing*, STOC '26, page 869–873, New York, NY, USA, 2026. Association for Computing Machinery. [pp. 3, 59]

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. [p. 4]

[KM93] Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993. [p. 4]

[KSS92] Michael J Kearns, Robert E Schapire, and Linda M Sellie. Toward efficient agnostic learning. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 341–352, 1992. [p. 5]

[LOH24] Lorenzo Leone, Salvatore F. E. Oliviero, and Alioscia Hamma. Learning t-doped stabilizer states. *Quantum*, 8:1361, May 2024. [p. 6]

[LS93] László Lovász and Miklós Simonovits. Random walks in a convex body and an improved volume algorithm. *Random Structures & Algorithms*, 4(4):359–412, 1993. [p. 55]

[Mon17] Ashley Montanaro. Learning stabilizer states by Bell sampling, 2017. [p. 6]

[MT25] Saeed Mehraban and Mehrdad Tahmasbi. Improved bounds for testing low stabilizer complexity states. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, page 1222–1233. Association for Computing Machinery, 2025. [p. 3]

[Ruz99] Imre Ruzsa. An analog of freiman's theorem in groups. *Astérisque*, 258(199):323–326, 1999. [p. 3]

[Sam07] Alex Samorodnitsky. Low-degree tests at large distances. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, page 506–515. Association for Computing Machinery, 2007. [pp. 3, 5, 51]

[Tao10] Terence Tao. Sumset and inverse sumset theory for shannon entropy. *Combinatorics, Probability and Computing*, 19(4):603–639, 2010. [p. 8]

[Tre04] Luca Trevisan. Some applications of coding theory in computational complexity. *Quaderni di Matematica*, 13:347–424, 2004. [p. 4]

[TW14] Madhur Tulsiani and Julia Wolf. Quadratic goldreich–levin theorems. *SIAM Journal on Computing*, 43(2):730–766, 2014. [pp. 5, 18, 48, 49, 51, 52]

[ZB11] Noga Zewi and Eli Ben-Sasson. From affine to two-source extractors via approximate duality. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 177–186. Association for Computing Machinery, 2011. [p. 3]

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il