

On Testing Finite Groups

Nader H. Bshouty

Henry and Marilyn Taub Faculty of Computer Science
Technion Israel Institute of Technology
bshouty@cs.technion.ac.il

Abstract

Let $G = [n]$ and let an unknown binary operation $* : G \times G \rightarrow G$ be given by oracle access to its multiplication table. The tester is given n explicitly as part of its input. A query chooses $x, y \in G$ and receives $x * y$. We study randomized adaptive two-sided property testers for deciding whether $(G, *)$ is a group. An input is considered far from the property if one must change at least an ϵ -fraction of the n^2 table entries in order to obtain any group operation on the same labeled set.

We prove three results. First, there is a tester using

$$O\left(n + \frac{1}{\epsilon}\right)$$

queries and $\tilde{O}(n + 1/\epsilon)$ time. In fact, this tester has one-sided error. This improves the Goldreich-Tauber tester [10], that uses $\tilde{O}(n/\epsilon)$ queries.

Second, if computational efficiency is ignored, a dictionary-and-locator construction combined with an action-consistency test on an expanding Cayley graph gives a tester using

$$\tilde{O}\left(\sqrt{\frac{n}{\epsilon}} + \frac{1}{\epsilon}\right)$$

queries. This is the first query complexity bound for this problem that is sublinear in n .

Third, there are infinitely many values of n for which every adaptive randomized two-sided tester needs

$$\Omega\left(n^{1/5} + \frac{1}{\epsilon}\right)$$

queries. This improves on the $\Omega(\log n)$ lower bound of Goldreich and Tauber. The lower bound uses two randomly relabeled abelian groups of almost equal orders, embeds them into multiplication tables on the same n -element set, and couples arbitrary adaptive query transcripts.

1 Introduction

A binary operation on an n -element set is specified by an $n \times n$ multiplication table. Property testing asks whether one can distinguish a group operation from an operation that is far from every group operation by inspecting only a small fraction of the n^2 table entries. Throughout the paper, the underlying labeled set is $G = [n] = \{1, 2, \dots, n\}$. One oracle query specifies an ordered pair $(x, y) \in G^2$ and receives the value $x * y$.

For another operation $\circ : G^2 \rightarrow G$, define (normalized Hamming) distance by

$$\text{dist}(*, \circ) := \frac{1}{n^2} \left| \{(x, y) \in G^2 : x * y \neq x \circ y\} \right|.$$

We say that $*$ is ϵ -far from the group property if $\text{dist}(*, \circ) \geq \epsilon$ for every group operation $\circ$ on the same set G .

A randomized tester has *two-sided error* if it may make an error on both kinds of inputs. We use the conventional success probability $2/3$: a genuine group must be accepted with probability at least¹ $2/3$, and an ϵ -far operation must be rejected with probability at least $2/3$. The tester is *adaptive*: the next query may depend on all previous answers. Throughout the paper, we assume that the tester is given $n = |G|$ explicitly as part of its input.

1.1 Our results

We prove three results on the query complexity of testing whether an unknown binary operation on $G = [n]$ defines a group.

Our first result is a computationally efficient tester.

Theorem 1.1. *There is a randomized adaptive one-sided ϵ -tester for the group property that uses*

$$O\left(n + \frac{1}{\epsilon}\right)$$

queries and runs in time $\tilde{O}(n + 1/\epsilon)$.

This improves the tester of Goldreich and Tauber [10], which uses $\tilde{O}(n/\epsilon)$ queries.

Our second result shows that the number of queries can be reduced further if computational efficiency is not required.

Theorem 1.2. *There is a randomized adaptive two-sided ϵ -tester for the group property that uses*

$$\tilde{O}\left(\sqrt{\frac{n}{\epsilon}} + \frac{1}{\epsilon}\right)$$

queries.

For every fixed $\epsilon > 0$, this bound is sublinear in n . This is the first sublinear query bound for testing the group property.

Our third result gives a lower bound that applies even to adaptive two-sided testers with unrestricted computation.

Theorem 1.3. *There are infinitely many values of n for which every randomized adaptive two-sided ϵ -tester for the group property requires*

$$\Omega\left(n^{1/5} + \frac{1}{\epsilon}\right)$$

queries.

This improves the $\Omega(\log n)$ lower bound of Goldreich and Tauber [10].

The two upper bounds use different amounts of reconstruction. The first algorithm spends $O(n)$ queries to build a compact representation of a group. It then uses only $O(1/\epsilon)$ additional queries to compare the input operation with the reconstructed group.

The second algorithm avoids reconstructing all labels. It builds a smaller random dictionary and uses random walks to locate arbitrary labels. It then tests whether the resulting partial map respects the action of a small set of random generators. Expansion of the candidate Cayley graph forces an action-consistent map to be nearly bijective. This gives the sublinear query bound

$$\tilde{O}\left(\sqrt{\frac{n}{\epsilon}} + \frac{1}{\epsilon}\right).$$

For the lower bound, we consider two randomly relabeled abelian groups of almost equal orders. We embed them into multiplication tables on the same n -element set and couple the transcripts of arbitrary adaptive testers. The construction shows that adaptive queries cannot make the problem arbitrarily easy. For infinitely many values of n , even when ϵ is a fixed constant, every adaptive two-sided tester requires $\Omega(n^{1/5})$ queries.

¹When the probability is 1, the tester is called a one-sided error tester.

1.2 Previous work

Local testing of algebraic structure goes back to the work of Blum, Luby, and Rubinfeld [3]. For multiplication tables, Ergün et al. [6] studied spot-checking associativity, cancellation, and the group property. Goldreich and Tauber [10] later revisited group testing, corrected a gap in an earlier group tester, and gave several new algorithms. Among their results are efficient one-sided testers with $\tilde{O}(n/\epsilon)$ complexity and a general adaptive lower bound $\Omega(\log n)$.

Efficient testing of special classes of groups has also been studied; see, for example, Friedl, Ivanyos, and Santha [9] and Bshouty [4]. Regularity testing for permutation groups was considered by Acciaro and Atkinson [1]. When the full table is available, exact verification was studied by Rajagopalan and Schulman [16]; Evra, Gadot, Klein, and Komargodski [7] gave an optimal deterministic algorithm linear in the input-table size.

Our proofs also use several standard external results as black boxes. Alon and Roichman [2], in a quantitative form due to Landau and Russell [13], show that $O(\log n)$ random generators typically give an expanding Cayley graph. Pyber’s enumeration theorem [15] bounds the number of groups of a given order. The lower bound uses a theorem of Ivanyos, Le Gall, and Yoshida [12] on approximate homomorphisms, Maynard’s bounded-gap theorem for primes [14], and a general $\Omega(1/\epsilon)$ lower bound of Bshouty and Goldreich [5] and Fischer [8].

1.3 Overview of the upper-bound techniques

The two upper bounds share one basic philosophy: instead of checking the group axioms directly on many triples, we try to construct a genuine group operation that explains most of the observed multiplication table. The difference is how much of that group operation we reconstruct.

The linear-query tester. A genuine group acts on itself by left multiplication: for every element $a \in G$, the map $L_a : x \mapsto a * x$ is a permutation of G . Moreover, the set of all such permutations forms the left-regular permutation representation of the group. We call this the left-regular *permutation action* (or just *action*) of G . The linear-query tester tries to reconstruct this permutation action without reading the entire $n \times n$ multiplication table.

We begin with the identity element and a small set $H \subseteq G$ for which we have already reconstructed the permutations corresponding to left multiplication by the elements of H . When H does not yet contain all elements of G , we choose an element $b \notin H$ and try to extend the known action to the larger set generated by H and b . The new elements are discovered in blocks of the form $H * t := \{h * t : h \in H\}$, which play the role of right cosets of H . For each such block, once one representative t is known, the already reconstructed action on H determines how the elements of H should act throughout the whole block. Thus we do not need to learn all products involving the new elements separately.

The extension procedure explores only the information needed to construct these new blocks and the corresponding permutations. If the known set H grows from h elements to k elements, the extension uses at most $2k - h$ queries. In every successful extension the known set at least doubles. Consequently, if $1 = h_0 < h_1 < \dots < h_r = n$ are the successive sizes of the reconstructed sets, then $h_{i+1} \geq 2h_i$, and the total reconstruction cost telescopes to $O(n)$. Thus we can construct a transitive² permutation action on all n elements using only a linear number of queries.

For a genuine group, the action obtained in this way is exactly the left-regular action³. For an arbitrary input operation, however, the same reconstruction procedure might produce a transitive permutation group that is not regular. Recall that an action is regular if, for every pair

² P is *transitive* if for every $x, y \in H$ there is a permutation $\phi \in P$ such that $\phi(x) = y$.

³The left-regular action of a group H is its action on itself by left multiplication: each $a \in G$ acts as the permutation $L_a(x) = a * x$. The action is called *regular* because for every $x, y \in G$ there is exactly one group element a such that $L_a(x) = y$.

$x, y \in G$, there is exactly one permutation in the action that sends x to y . Regularity is essential because a regular permutation action determines a unique group structure on the underlying set.

We therefore run a randomized regularity test after every successful extension. The test checks whether the permutation group constructed so far has a nontrivial element fixing the distinguished identity element. If such an element is found, the input cannot be following the regular action of a group and the tester rejects. Since the known set at least doubles after each extension, there are only $O(\log n)$ extension stages, and the total cost of all regularity tests remains within the claimed linear-query bound.

If the reconstruction reaches all n elements and the resulting action is regular, then this action defines an actual group operation $\circ$ on G . Namely, for each $x \in G$ there is a unique permutation L_x in the regular action sending the identity to x , and we define

$$x \circ y := L_x(y).$$

Thus the reconstruction produces a genuine group operation $\circ$ that can be compared with the original input operation $*$.

It remains to perform this comparison efficiently. Explicitly constructing the entire multiplication table of $\circ$ would require n^2 time. Instead, we choose a short random set of generators for the reconstructed group. With high probability, the corresponding Cayley graph is an expander, which yields a succinct representation that allows us to evaluate $x \circ y$ in $O(\log n)$ time without making any further queries to the input table. We then choose $O(1/\epsilon)$ independent random pairs $(x, y) \in G^2$, query $x * y$, and compare it with the reconstructed value $x \circ y$. If the input operation is ϵ -far from every group operation, then in particular it is ϵ -far from $\circ$, so a random pair detects a disagreement with probability at least ϵ .

Combining the $O(n)$ reconstruction queries with the $O(1/\epsilon)$ final comparison queries gives

$$O\left(n + \frac{1}{\epsilon}\right)$$

queries. The reconstruction and succinct evaluation procedures can be implemented in total time $\tilde{O}(n + 1/\epsilon)$.

The sublinear-query tester. To use fewer than n queries, we do not try to reconstruct the whole multiplication table. We first choose $k = O(\log n)$ random generator candidates $u_1, \dots, u_k \in G$. For each $i \in [k]$, we also compute an element u_{-i} that is intended to serve as the inverse of u_i . Using these elements, we perform a random walk in G of length $m = \tilde{O}(\sqrt{n/\epsilon})$. Thus, starting from a random element d_0 , we repeatedly multiply on the right by a randomly chosen generator candidate or candidate inverse, obtaining $d_r = d_{r-1} * u_{\xi_r}$, $1 \leq r \leq m$. The set $D = \{d_0, d_1, \dots, d_m\}$ of elements visited by this walk will serve as a small *dictionary*.

Since computational efficiency is not required, we enumerate one representative H from each isomorphism class of abstract groups of order n , together with all possible interpretations of the sampled elements inside H . A marked candidate has the form $\mathcal{H} = (H, \rho, a_1, \dots, a_k)$, where $\rho \in H$ is the proposed image of d_0 , and $a_i \in H$ is the proposed image of the sampled generator u_i . The same sequence of generator choices $\xi_1, \dots, \xi_m$ defines a walk in H , $h_0 = \rho$, $h_r = h_{r-1} a_{\xi_r}$. We retain only candidates for which the two walks have exactly the same coincidences:

$$d_r = d_s \iff h_r = h_s \quad (0 \leq r, s \leq m).$$

Consequently, for every retained candidate there is a well-defined injective map from the dictionary into H , given by $d_r \mapsto h_r$. Thus the dictionary gives us a small set of input elements whose proposed images in each retained candidate group are already known.

We next use a second, common random walk as a *locator*. Starting from an arbitrary element $x \in G$, we repeatedly multiply on the right by the same sequence of randomly chosen generator

candidates and stop when the walk first reaches an element of the dictionary. If the first hit is $d \in D$, then, for each retained candidate $\mathcal{H}$, the known image of d , together with the generator steps taken along the walk, determines a proposed image

$$\Psi_{\mathcal{H}}(x) \in H.$$

If the walk does not hit the dictionary, then $\Psi_{\mathcal{H}}(x)$ is left undefined. On a genuine group, the randomly sampled generators give an expanding Cayley graph with high probability, and this expansion implies that a locator of length about $\tilde{O}(n/m)$ hits the dictionary from almost every starting element. Thus, for the true candidate, $\Psi_{\mathcal{H}}$ is defined on almost all of G and agrees there with the true isomorphism.

For soundness, however, the input operation need not be a group, and a retained candidate may induce a partial map with many collisions. The main task is therefore to determine whether $\Psi_{\mathcal{H}} : G \dashrightarrow H$ is close to a bijection. We first discard candidates $\mathcal{H}$ whose chosen generators do not give a Cayley graph of sufficiently large conductance. For each remaining candidate, we test local action-consistency relations: For a random $x \in G$ and a random signed generator u_j , we check that $\Psi_{\mathcal{H}}(x * u_j) = \Psi_{\mathcal{H}}(x) a_j$ whenever the two values of $\Psi_{\mathcal{H}}$ are defined, and we also check that the proposed inverse step satisfies $(x * u_j) * u_{-j} = x$. We show that if the tests succeed, then the image of $\Psi_{\mathcal{H}}$ contains almost all elements of H ; equivalently, $\Psi_{\mathcal{H}}$ is close to a bijection.

Finally, we test whether the surviving partial map approximately preserves multiplication. For random $x, y \in G$, we compare

$$\Psi_{\mathcal{H}}(x * y) \quad \text{with} \quad \Psi_{\mathcal{H}}(x) \Psi_{\mathcal{H}}(y).$$

A partial map that is both close to a bijection and approximately multiplicative allows us to transport the multiplication of H back to G , producing a genuine group operation that is close to the input operation. Hence an input that is ϵ -far from every group operation cannot have a candidate that passes both the action-consistency and product tests.

The dictionary costs m queries to build, while locating a random element costs about $O(n/m)$ queries. Balancing these two costs gives $m = \tilde{\Theta}(\sqrt{n/\epsilon})$ and leads to the query complexity

$$\tilde{O}\left(\sqrt{\frac{n}{\epsilon}} + \frac{1}{\epsilon}\right).$$

The same tester applies, without changing its query complexity, to any class $\mathcal{C}$ of finite groups that is closed under isomorphism. It suffices to enumerate only candidate groups $H \in \mathcal{C}$. Completeness is unchanged, because an input group belonging to $\mathcal{C}$ is isomorphic to one of these candidates. For soundness, if a candidate $H \in \mathcal{C}$ passes both tests, the argument above produces a group operation close to the input that is isomorphic to H , and therefore also belongs to $\mathcal{C}$.

1.4 Overview of the lower-bound technique

The adaptive lower bound is based on two finite abelian groups that have almost the same order and satisfy the same low-complexity algebraic identities, but are not isomorphic. We choose consecutive primes $p < q$ with bounded gap, so that $q - p = O(1)$, and define

$$H := \mathbb{F}_p^3 \times \mathbb{F}_q^2, \quad K := \mathbb{F}_p^2 \times \mathbb{F}_q^3.$$

Their orders are $|H| = p^3 q^2$ and $|K| = p^2 q^3$. Thus $|H|/|K| = p/q = 1 - O(1/p)$, so the two groups have almost the same size. At the same time, both groups have *exponent* pq : That is, every element x satisfies $pqx = 0$. This common integer is important because it makes short symbolic computations in the two groups look the same.

Put $m := |H|$, and $n := |K|$. The YES distribution is obtained by choosing a uniformly random bijection $\pi_K : K \rightarrow [n]$ and transporting the group operation of K to $[n]$. Thus every

YES instance is simply a randomly relabeled copy of the group K . For the NO distribution, choose a uniformly random subset $S \subseteq [n]$, $|S| = m$, and a uniformly random bijection $\pi_H : H \rightarrow S$. On $S \times S$ we place the multiplication table of H , using this random labeling. Outside $S \times S$ the table can be completed by any fixed rule. Thus almost all elements of $[n]$ belong to the hidden copy of H , because $|S|/n = p/q = 1 - O(1/p)$. Only an $O(1/p)$ fraction of the visible labels $[n]$ lie outside the hidden copy.

The NO instances are nevertheless far from every group operation on $[n]$. The reason is that $|H| = p^3 q^2$ does not divide $n = |K| = p^2 q^3$. Hence, by Lagrange's theorem, H cannot be isomorphic to a subgroup of any group of order n . The overlap theorem of Ivanyos–Le Gall–Yoshida then implies that no n -element group can agree with the hidden multiplication table of H on more than a constant fraction of the pairs in $S \times S$. Since $|S|/n = p/q = 1 - o(1)$, the set $S \times S$ itself occupies a constant fraction tending to one of the whole table. Therefore every NO instance is a constant distance from every group operation on $[n]$.

It remains to explain why a tester making only a small number of adaptive queries cannot distinguish the two distributions. The main idea is to reveal the random labels only when they are needed. Instead of fixing the entire random relabeling in advance, we expose it lazily as the tester makes queries.

Every new visible label introduced by the tester is assigned a formal symbolic value. If, for example, the tester has already encountered formal values $X_1, \dots, X_s$ and asks for the product of two labels represented symbolically by $\alpha(X_1, \dots, X_s)$ and $\beta(X_1, \dots, X_s)$, then the answer is represented symbolically by

$$\alpha(X_1, \dots, X_s) + \beta(X_1, \dots, X_s),$$

because both H and K are written additively. Thus the tester first sees a purely symbolic transcript consisting of linear combinations of the formal variables.

The crucial point is that the same symbolic transcript can be realized inside either H or K . Both groups are $\mathbb{Z}_{pq}$ -modules⁴ and therefore decompose into their p -primary and q -primary components, which are vector spaces over $\mathbb{F}_p$ and $\mathbb{F}_q$, respectively. Therefore all formal linear relations that follow only from the exponent are the same in the two groups. As long as no unexpected equality occurs between two distinct formal expressions, the tester sees exactly the same pattern of new and repeated labels in the two experiments.

There are only two main ways in which the coupling between the YES and NO transcripts can fail: First, in the NO distribution the tester may explicitly mention a visible label outside the hidden set S . Since only $\rho := (n - m)/n = O(1/p)$ of the labels lie outside S , and at most $O(t)$ fresh visible labels can be introduced during t queries, the probability of this event is $O(t\rho) = O(t/p)$.

Second, two formally distinct expressions may accidentally evaluate to the same group element. Such an accidental equality corresponds to a nontrivial linear relation among the exposed symbolic variables. For the particular pair $H = \mathbb{F}_p^3 \times \mathbb{F}_q^2$, and $K = \mathbb{F}_p^2 \times \mathbb{F}_q^3$, the smallest relevant primary component has size p^2 . Hence any fixed nontrivial formal equality occurs with probability at most $O(1/p^2)$. Since a transcript containing t queries creates only $O(t)$ symbolic expressions, there are only $O(t^2)$ pairs of expressions that might collide. A union bound therefore gives total accidental-collision probability $O(t^2/p^2)$.

Combining the two sources of error, the total variation distance between the transcript distributions in the YES and NO experiments is at most

$$O\left(\frac{t}{p} + \frac{t^2}{p^2}\right).$$

⁴A $\mathbb{Z}_{pq}$ -module is analogous to a vector space, except that the scalars come from the ring $\mathbb{Z}_{pq}$ rather than from a field.

Thus, if $t = cp$ for a sufficiently small absolute constant $c > 0$, this total variation distance remains bounded away from 1. In particular, no adaptive tester using only cp queries can distinguish the two distributions with constant advantage. Yao's minimax principle therefore gives an adaptive lower bound $\Omega(p)$.

Finally, because $q - p = O(1)$, we have $q = \Theta(p)$ and therefore $n = |K| = p^2 q^3 = \Theta(p^5)$. Hence $p = \Theta(n^{1/5})$, which yields the lower bound $\Omega(n^{1/5})$. Combining this with the general $\Omega(1/\epsilon)$ lower bound gives

$$\Omega\left(n^{1/5} + \frac{1}{\epsilon}\right)$$

throughout the stated range of ϵ .

The same construction applies more generally to every class $\mathcal{C}$ of finite groups that is closed under isomorphism and contains $H = \mathbb{F}_p^3 \times \mathbb{F}_q^2$ and $K = \mathbb{F}_p^2 \times \mathbb{F}_q^3$. The YES instances are random relabelings of K , and therefore belong to $\mathcal{C}$. The NO instances are constant-far from every group operation, and hence in particular are constant-far from every group in $\mathcal{C}$. Thus the same lower bound holds for testing membership in $\mathcal{C}$.

2 Preliminaries and black boxes

2.1 The adaptive two-sided testing model

Let $G = [n]$, and let

$$\mathcal{G}_n := \{\circ : G \times G \rightarrow G : (G, \circ) \text{ is a group}\}.$$

The input is an arbitrary binary operation $* : G \times G \rightarrow G$. The tester is given n and ϵ , and has oracle access to the multiplication table of $*$. A query consists of a pair $(x, y) \in G^2$, and the oracle returns $x * y$. No further promise is made about the input operation.

For two operations $*, \circ : G^2 \rightarrow G$, define their normalized Hamming distance by

$$\text{dist}(*, \circ) := \frac{1}{n^2} \left| \left\{ (x, y) \in G^2 : x * y \neq x \circ y \right\} \right|.$$

We also write

$$\text{dist}(*, \mathcal{G}_n) := \min_{\circ \in \mathcal{G}_n} \text{dist}(*, \circ).$$

Thus, $*$ is ϵ -far from the group property if $\text{dist}(*, \mathcal{G}_n) \geq \epsilon$.

Definition 2.1 (Adaptive two-sided tester). A randomized oracle algorithm T is an adaptive two-sided ϵ -tester for the group property if it satisfies the following conditions. For an input operation $*$, let $T(n, \epsilon)$ denote the output of the tester when given oracle access to $*$. Then

$$\begin{aligned} * \in \mathcal{G}_n &\implies \Pr[T(n, \epsilon) = \text{accept}] \geq \frac{2}{3}, \\ \text{dist}(*, \mathcal{G}_n) \geq \epsilon &\implies \Pr[T(n, \epsilon) = \text{reject}] \geq \frac{2}{3}. \end{aligned}$$

The probability is over the internal randomness of T . No guarantee is required when $0 < \text{dist}(*, \mathcal{G}_n) < \epsilon$.

The tester is adaptive: each query may depend on n , ϵ , its internal randomness, and the complete sequence of previous queries and answers. Its query complexity is the maximum number of oracle queries made over all input operations and all outcomes of its internal randomness. Computation between queries and after the final query is unrestricted.

The query lower bound in this paper applies to adaptive two-sided testers in this model.

Distances between two multiplication tables are integer multiples of $1/n^2$. It is therefore convenient to define $\bar{\epsilon} := \max\{\epsilon, 1/n^2\}$. If an operation is ϵ -far from every group, then it is also $\bar{\epsilon}$ -far.

2.2 Permutation actions

A *permutation* of a finite set X is a bijection $X \rightarrow X$. $\text{Sym}(X)$ is the *symmetric group* on X , i.e. the group of all permutations of X . A group P of permutations of X naturally acts on X : each $g \in P$ sends $x \in X$ to $g(x)$. This action is called *transitive* if, for every $x, y \in X$, there is some $g \in P$ such that $g(x) = y$. Fix $e \in X$. The *stabilizer* of e is

$$P_e := \{g \in P : g(e) = e\}.$$

A transitive action is called *regular* if $P_e = \{1\}$ where 1 denotes the identity permutation. Equivalently, for every $x, y \in X$ there is exactly one permutation in P sending x to y .

The following elementary observation is central to the first upper bound.

Lemma 2.2 (A regular action gives a group). *Let $P \leq \text{Sym}(X)$ act regularly on X , and fix $e \in X$. For each $x \in X$, let p_x be the unique permutation in P satisfying $p_x(e) = x$. Define*

$$x \circ y := p_x(y).$$

Then $(X, \circ)$ is a group with identity e , and the map $x \mapsto p_x$ is an isomorphism from $(X, \circ)$ to P .

Proof. For $x, y \in X$, $p_x p_y(e) = p_x(y) = x \circ y$. By regularity, $p_x p_y$ is the unique permutation sending e to $x \circ y$. Hence $p_x p_y = p_{x \circ y}$. Thus $x \mapsto p_x$ is a bijective homomorphism onto the group P . All group axioms for $\circ$ follow immediately. $\square$

2.3 Random Cayley graphs

Let H be a finite group. A multiset Σ of elements of H is called *symmetric* if every element and its inverse occur with the same multiplicity. The *right Cayley multigraph* associated with H and Σ has vertex set H , and for every $x \in H$ and every $a \in \Sigma$ it has an edge $x \rightarrow xa$. If the elements of Σ generate H , then this graph is connected. The *normalized transition matrix* P of the Cayley multigraph describes one step of the following random walk: from x , choose an element $a \in \Sigma$ uniformly, counting multiplicities, and move to xa . That is, $P(x, y) := |\{a \in \Sigma : xa = y\}|/|\Sigma|$.

We use the following form of the Alon–Roichman theorem. Alon and Roichman [2] proved that $O(\log |H|)$ independent uniform random elements give a Cayley graph with a constant spectral gap⁵ The tail estimate in the form used here is given explicitly by Landau and Russell [13, Theorem 2]. The definitions and the short deduction from their theorem are given in Appendix A.

Theorem 2.3 (Random Cayley expansion). *For every fixed $0 < \lambda < 1$, there is a constant $c_{\text{AR}} = c_{\text{AR}}(\lambda) > 0$ such that the following holds.*

Let H be any group of order n , and choose $k = \lceil c_{\text{AR}} \log n \rceil$ independent uniform elements $a_1, \dots, a_k \in H$. Form the symmetric multiset $\Sigma = \{a_1, a_1^{-1}, \dots, a_k, a_k^{-1}\}$, and let P be the normalized transition matrix of the corresponding right Cayley multigraph. Then, with probability at least $99/100$,⁶

$$\left\| P - \frac{J}{n} \right\|_{2 \rightarrow 2} \leq \lambda,$$

where J is the $n \times n$ all-ones matrix.

Fix from now on an absolute constant $0 < \lambda_0 \leq 1/3000$. and set $c_{\text{AR}} := c_{\text{AR}}(\lambda_0)$. Every later invocation of Theorem 2.3 uses this fixed value λ_0 . Thus the desired spectral accuracy is chosen first, and the number of random elements is chosen afterward.

We also use the following simpler consequence.

⁵The eigenvalue 1 is the trivial eigenvalue of P . If $1 = \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ are the eigenvalues of P , then the spectral gap is $1 - \lambda_2$.

⁶Here the notation $\left\| P - \frac{J}{n} \right\|_{2 \rightarrow 2}$ measures the largest absolute value of a nontrivial eigenvalue of P . A precise explanation is given in Appendix A.

Corollary 2.4 (Short random generators). *There are absolute constants $c_0, c_1 > 0$ such that, for every group H of order n , if $k = \lceil c_0 \log n \rceil$ independent uniform elements of H are chosen, then with probability at least $2/3$, every element of H is a product of at most $c_1 \log n$ of the sampled elements and their inverses.*

The proof is also included in Appendix A. Thus readers who do not know spectral graph theory may use Theorem 2.3 and Corollary 2.4 simply as black boxes.

2.4 Counting groups

We also use Pyber’s enumeration theorem [15] only through the following simple consequence.

Theorem 2.5 (Pyber). *Let $g(n)$ be the number of isomorphism classes of groups of order $n \geq 2$. There is an absolute constant C such that*

$$g(n) \leq \exp(C(\log n)^3).$$

2.5 Two black boxes for the lower bound

The first is an overlap theorem for group multiplication.

Theorem 2.6 (Ivanyos–Le Gall–Yoshida [12]). *Let A and B be finite groups with $|A| \leq |B|$. If A is not isomorphic to a subgroup of B , then every injection $f : A \rightarrow B$ satisfies*

$$\Pr_{x,y \in A} [f(xy) = f(x)f(y)] \leq \frac{7}{9}.$$

The second supplies two very close primes infinitely often.

Theorem 2.7 (Consequence of Maynard [14]). *As a corollary of Maynard’s bounded-gap theorem, there are infinitely many pairs of consecutive primes $p < q$ with*

$$q - p \leq 600.$$

3 An $O(n + 1/\epsilon)$ adaptive tester

In this section we give a computationally efficient one-sided error tester.

Theorem 3.1 (Linear-query adaptive tester). *For every $n \geq 4$ and $0 < \epsilon < 1/2$, there is a randomized adaptive tester which*

- (i) *accepts every group with probability 1;*
- (ii) *rejects every ϵ -far operation with probability at least $2/3$;*

makes $O(n + 1/\epsilon)$ oracle queries and runs in $O(n \log n \cdot \log \log n + (\log n)/\epsilon)$ time.

3.1 Growing a candidate group in blocks

Let G be the finite set on which the input operation $*$: $G \times G \rightarrow G$ is defined, and write $n := |G|$. Suppose that we have already constructed a subset $H \subseteq G$, together with a set Γ_H of permutations of H . Write $P_H := \langle \Gamma_H \rangle \leq \text{Sym}(H)$. We assume that P_H acts transitively on H . We also fix a distinguished point $e \in H$. Write $h := |H|$.

The set H should be thought of as a candidate subgroup, and the permutations in Γ_H as candidate left translations⁷. At this stage, however, we do not assume that the input operation $*$ is associative, or even that H is actually a group.

Fix an element $b \in G \setminus H$ that we wish to adjoin to H . The element b remains fixed throughout one call of the procedure below. Define $\beta(x) := b * x$. Thus β is the map given by left multiplication by b . The algorithm does not know the whole map β in advance. Whenever it needs the value $\beta(x)$, it learns it by querying the table entry $b * x$.

The idea is to discover a larger set by dividing it into disjoint blocks, each having $h = |H|$ elements. The first block is simply $C_e := H$. For this initial block, define $\mu_e : H \rightarrow C_e$ by $\mu_e(c) := c$. Thus μ_e is the identity map on H .

Every later block will have a representative $t \in G$. For such a representative, the map $\mu_t : H \rightarrow G$ is learned from the input table by querying $\mu_t(c) := c * t$, where $c \in H$. If the resulting values pass the checks in the algorithm, they form a new block $C_t := \mu_t(H)$. If $(G, *)$ is a group and $H \leq G$, then $C_t = \{c * t : c \in H\} = Ht$, so C_t is the right coset of H determined by t .

The following procedure performs the extension.

Algorithm 1: Explore-Extension

Require: A set $H \subseteq G$, permutation generators $\Gamma_H \subseteq \text{Sym}(H)$, a distinguished point $e \in H$, and a fixed element $b \in G \setminus H$.

Ensure: Either REJECT, or a larger set $K \supseteq H$ together with permutation generators $\Gamma_K \subseteq \text{Sym}(K)$.

```

1:  $h \leftarrow |H|$ ,  $D \leftarrow H$ , and  $\mathcal{T} \leftarrow \{e\}$ .
2:  $C_e \leftarrow H$ , and define  $\mu_e(c) \leftarrow c$  for every  $c \in H$ .
3: Put every element of  $H$  into a queue  $Q$ .
4: Initialize  $\beta$  as an empty partial map.
5: while  $Q$  is nonempty do
6:   Remove one point  $x$  from  $Q$ .
7:   Query  $\beta(x) \leftarrow b * x$ .
8:   if  $x = e$  and  $\beta(e) \neq b$ , return REJECT.
9:    $t \leftarrow \beta(x)$ .
10:  if  $t \notin D$  then  $\triangleright t$  represents a new block
11:    for each  $c \in H$  do
12:      Query  $c * t$  and set  $\mu_t(c) \leftarrow c * t$ .
13:    end for
14:     $C_t \leftarrow \{\mu_t(c) : c \in H\}$ .
15:    if  $\mu_t(e) \neq t$ , or the values  $\mu_t(c)$  are not all distinct, or  $C_t \cap D \neq \emptyset$  then
16:      return REJECT.
17:    end if
18:     $\mathcal{T} \leftarrow \mathcal{T} \cup \{t\}$ .  $D \leftarrow D \cup C_t$ .
19:    Put every element of  $C_t$  into  $Q$ .
20:  end if
21: end while
22: if  $\beta(x) \notin D$  for some  $x \in D$ , or the values  $\beta(x)$ ,  $x \in D$ , are not all distinct then
23:  return REJECT.
24: end if
25: for each  $\gamma \in \Gamma_H$  do
26:   Define

```

$$\hat{\gamma}(\mu_t(c)) \leftarrow \mu_t(\gamma(c)), \quad t \in \mathcal{T}, \quad c \in H.$$

⁷In a genuine group, left multiplication by $a \in H$ is the permutation $L_a(x) = a * x$. The permutations in Γ_H are intended to play this role.

27: **end for**

28: $K \leftarrow D$.

29:

$$\Gamma_K \leftarrow \{\hat{\gamma} : \gamma \in \Gamma_H\} \cup \{\beta\}.$$

30: **return** (K, Γ_K) .

Let us explain the objects appearing in the algorithm. The *discovered set* $D \subseteq G$ consists of all points found so far. The set $\mathcal{T} \subseteq D$ contains one representative t for each block. For every $t \in \mathcal{T}$, the corresponding block is $C_t = \mu_t(H)$, and, if the algorithm has not rejected, $\mu_t : H \rightarrow C_t$ is a bijection.

The queue Q contains precisely those discovered points for which the value $\beta(x) = b * x$ has not yet been queried. Since every new block is required to be disjoint from the previously discovered set, every point enters the queue exactly once.

Notice also that b is fixed throughout the entire procedure. Thus the queries $\beta(x) = b * x$ all lie in the single row of the multiplication table corresponding to b . The point x varies over the discovered set; b does not vary.

The only other multiplication-table queries are those used to build a new block: $\mu_t(c) = c * t$, where $c \in H$. Once the values of μ_t have been learned, extending an old stored permutation $\gamma \in \Gamma_H$ by $\hat{\gamma}(\mu_t(c)) = \mu_t(\gamma(c))$ requires no further query to the input operation.

The next lemmas explain what the procedure guarantees. Unless a lemma explicitly assumes that $(G, *)$ is a group, no group law is assumed for the input table.

Lemma 3.2 (Block invariant). *If EXPLORE-EXTENSION has not rejected, then at every stage*

$$D = \bigsqcup_{t \in \mathcal{T}} C_t,$$

where $C_t = \mu_t(H)$, and every map $\mu_t : H \rightarrow C_t$ is a bijection satisfying $\mu_t(e) = t$.

Proof. Initially, $D = H = C_e$, $\mathcal{T} = \{e\}$, and $\mu_e : H \rightarrow H$, where $\mu_e(c) = c$, is a bijection satisfying $\mu_e(e) = e$.

Suppose that a new block with representative t is added. The procedure in step 15 checks that the $h = |H|$ values $\mu_t(c)$, $c \in H$, are all distinct. Hence $\mu_t : H \rightarrow C_t$ is a bijection. The procedure also checks $\mu_t(e) = t$ and $C_t \cap D = \emptyset$. Thus the new block is disjoint from every previously constructed block. The result follows by induction over the blocks that are created. $\square$

Lemma 3.3 (The old action embeds). *For every $q \in P_H$, define a map $\hat{q} : D \rightarrow D$ block by block by $\hat{q}|_{C_t} := \mu_t q \mu_t^{-1}$, $t \in \mathcal{T}$.*

Then $\hat{q}|_{C_t}$ is a permutation of C_t , $\hat{q}$ is a permutation of D , and $q \mapsto \hat{q}$ is an injective homomorphism

$$P_H \hookrightarrow \text{Sym}(D).$$

For $q = \gamma \in \Gamma_H$, this is exactly the permutation $\hat{\gamma}$ constructed by EXPLORE-EXTENSION.

Proof. Fix $t \in \mathcal{T}$. Since, by Lemma 3.2, $\mu_t : H \rightarrow C_t$ is a bijection and q is a permutation of H , the map $\mu_t q \mu_t^{-1} : C_t \rightarrow C_t$ is a permutation of C_t . The blocks C_t are pairwise disjoint and cover D , so the blockwise definitions together give a permutation of D .

For $q_1, q_2 \in P_H$, on every block C_t ,

$$\widehat{q_1 q_2} = \mu_t q_1 q_2 \mu_t^{-1} = (\mu_t q_1 \mu_t^{-1})(\mu_t q_2 \mu_t^{-1}) = \hat{q}_1 \hat{q}_2.$$

Hence $q \mapsto \hat{q}$ is a homomorphism.

Now, on the initial block $H = C_e$, we have $\mu_e = \text{id}_H$. Therefore $\hat{q}|_H = q$. Thus $\hat{q}$ determines q , and the homomorphism is injective.

Finally, let $q = \gamma \in \Gamma_H$. For any $t \in \mathcal{T}$ and any $c \in H$,

$$\hat{q}(\mu_t(c)) = (\mu_t \gamma \mu_t^{-1})(\mu_t(c)) = \mu_t(\gamma(c)).$$

This is exactly the definition of $\hat{\gamma}$ in EXPLORE-EXTENSION. Hence, for $q = \gamma \in \Gamma_H$, the permutation $\hat{q}$ defined in the lemma is precisely the permutation $\hat{\gamma}$ constructed by the algorithm. $\square$

If the procedure returns K , define

$$P_K := \langle \Gamma_K \rangle \leq \text{Sym}(K).$$

Lemma 3.4 (Transitivity and doubling). *If EXPLORE-EXTENSION returns K , then P_K acts transitively on K , and*

$$|K| \geq 2|H|.$$

Proof. We prove by induction that every point ever placed in the queue is reachable from e .⁸ This is true initially, since the queue contains exactly the points of H , and P_H acts transitively on H .

Now suppose that the procedure removes a point x from the queue. By the induction hypothesis, x is reachable from e . If a new block C_t is created while processing x , then, by definition, $t = \beta(x)$. Since $\beta \in \Gamma_K \subseteq P_K$, the representative t is also reachable from e .

Take any point of the new block. It has the form $\mu_t(c)$ for some $c \in H$. Since P_H acts transitively on H , there is some $q_c \in P_H$ such that $q_c(e) = c$. By Lemma 3.3, its extension $\hat{q}_c$ belongs to P_K . Since $t = \mu_t(e)$, we have

$$\hat{q}_c(t) = \hat{q}_c(\mu_t(e)) = \mu_t(q_c(e)) = \mu_t(c).$$

Thus every point of C_t is reachable from t , and hence from e . Since these are the only new points added to the queue, the induction invariant is preserved.

Therefore every point of K is reachable from e , and hence P_K acts transitively on K .

For the doubling statement, the procedure requires $\beta(e) = b$. Since $b \notin H$, the value $\beta(e) = b$ lies outside the initial block H . Consequently the algorithm creates at least one additional block. By the block invariant, every block has exactly $|H| = h$ elements. Hence $|K| \geq 2|H|$. $\square$

Lemma 3.5 (Linear query cost). *Suppose that an extension starts with $|H| = h$ and successfully returns a set K of size $|K| = k$. Then EXPLORE-EXTENSION makes at most $2k - h$ queries to the input multiplication table.*

If the extension rejects, it makes $O(n)$ queries, where $n = |G|$.

Proof. In a successful extension, every discovered point enters the queue exactly once. Therefore the procedure makes exactly one query $b * x$ for every $x \in K$. This gives k queries.

By the block invariant, K is a disjoint union of blocks, each of size h . Hence there are k/h blocks. The initial block $C_e = H$ requires no multiplication-table queries to construct. Every later block requires at most h queries, $c * t$, where $c \in H$. Thus the total number of block-construction queries is

$$\left(\frac{k}{h} - 1\right) h = k - h.$$

Consequently the total query count is at most $k + (k - h) = 2k - h$.

Notice that extending the stored permutations $\gamma \in \Gamma_H$ does not add any multiplication-table queries. Indeed, $\hat{\gamma}(\mu_t(c)) = \mu_t(\gamma(c))$ is computed entirely from the stored value $\gamma(c)$ and the

⁸Since P_K is a group, if x is reachable from e , then e is reachable from x by the inverse permutation. Hence, if both x and y are reachable from e , then y is reachable from x .

already known map μ_t . This computation contributes to the running time, but not to the oracle-query count.

Now suppose that the extension rejects. At most n points of G can be successfully discovered, so there are at most n queries of the form $b * x$. The successfully constructed blocks are pairwise disjoint, so together they require at most n block-construction queries. A final block on which the procedure rejects requires at most $h \leq n$ additional queries. Hence a failed extension makes $O(n)$ queries. $\square$

The preceding lemmas did not assume that the input table is a group. The next lemma explains what happens when the input operation really is a group operation.

Lemma 3.6 (Honest extension). *Suppose that $(G, *)$ is a group with identity e , that $H \leq G$ is a subgroup, and that P_H is the true left-regular action of H on itself. Thus, for every $a \in H$, the corresponding left translation is $L_a(c) := a * c$.*

Then, for every $b \in G \setminus H$, EXPLORE-EXTENSION does not reject. It returns⁹

$$K = \langle H, b \rangle_*,$$

the subgroup of G generated by H and b , and every returned permutation is the restriction to K of a true left translation. In particular, the returned action is regular.

Proof. Consider a new representative t . The block constructed by the procedure is

$$C_t = \{\mu_t(c) : c \in H\} = \{c * t : c \in H\} = Ht,$$

which is the right coset of H containing t . Because $(G, *)$ is a group, right cancellation implies that the map $c \mapsto c * t$ is injective. Since e is the identity, $\mu_t(e) = e * t = t$. Also, two right cosets of H are either equal or disjoint. At every stage, the discovered set D is a union of right cosets of H . A new block is proposed only when $t \notin D$. Therefore the right coset Ht is different from every previously discovered coset, and hence is disjoint from D . Thus all the block checks pass.

Now let an old generator $\gamma \in \Gamma_H$ be left multiplication by some $a \in H$: $\gamma(c) = a * c$. Take $z = \mu_t(c) = c * t$. Then

$$\hat{\gamma}(z) = \mu_t(\gamma(c)) = \mu_t(a * c) = (a * c) * t = a * (c * t) = a * z.$$

Thus $\hat{\gamma}$ is true left multiplication by a on every discovered block.

By definition, $\beta(z) = b * z$, so β is true left multiplication by b . In particular, left multiplication by b is a permutation. Hence the final permutation check on β also succeeds.

Every point ever discovered belongs to $\langle H, b \rangle_*$. Indeed, this is true initially for the points of H . If $x \in \langle H, b \rangle_*$, then $\beta(x) = b * x$ also lies in this subgroup. If a new representative $t \in \langle H, b \rangle_*$ is discovered, then every point of its block has the form $c * t$, where $c \in H$, and hence also belongs to $\langle H, b \rangle_*$. Therefore $K \subseteq \langle H, b \rangle_*$.

Conversely, when the procedure terminates, K is invariant under left multiplication by every element of H . Indeed, each block is a right coset Ht , and left multiplication by an element of H maps Ht to itself. The set K is also invariant under $L_b = \beta$. Since $\beta : K \rightarrow K$ is a permutation, K is invariant under its inverse $\beta^{-1} = L_{b^{-1}}$ as well.

Thus K , which contains e , contains every element that can be obtained from e by repeatedly applying left multiplication by elements of $H \cup \{b, b^{-1}\}$. This set is exactly $\langle H, b \rangle_*$.

The returned permutation group is consequently the true left-regular action of this subgroup, and hence it is regular. $\square$

⁹One could simply write $\langle H, b \rangle$. We write $\langle H, b \rangle_*$ only to emphasize that the subgroup is generated using the original operation $*$, since later we also introduce induced group operations such as $\circ_H$ and $\circ_K$.

Even when the input table is not a group, regularity of the returned action is enough to define and certify a genuine group extension.

By Lemma 2.2, a regular permutation action on a set with distinguished point e induces a group operation on that set.

Lemma 3.7 (Certified extension). *Suppose that the old action P_H on H is regular and that EXPLORE-EXTENSION returns a set K for which the new action $P_K = \langle \Gamma_K \rangle$ is also regular.*

Let $\circ_H$ be the group operation on H induced by the old regular action, and let $\circ_K$ be the group operation on K induced by the new regular action. Then

$$H \leq (K, \circ_K),$$

the two operations agree on $H \times H$,

$$x \circ_H y = x \circ_K y, \quad x, y \in H,$$

and

$$K = \langle H, b \rangle_{\circ_K}.$$

Proof. By Lemma 3.3, the old permutation group P_H embeds into P_K . Fix $h \in H$. Since the old action is regular, there is a unique permutation $q_h \in P_H$ such that $q_h(e) = h$. Its extension $\hat{q}_h \in P_K$ still sends e to h , because on the initial block H it agrees with q_h .

Since the new action is also regular, there is exactly one permutation in P_K that sends e to h . Therefore $\hat{q}_h$ is precisely the left translation corresponding to h in the group $(K, \circ_K)$.

Hence, for $h_1, h_2 \in H$,

$$h_1 \circ_K h_2 = \hat{q}_{h_1}(h_2) = q_{h_1}(h_2) = h_1 \circ_H h_2.$$

Thus the operations $\circ_H$ and $\circ_K$ agree on $H \times H$, and consequently $H \leq (K, \circ_K)$.

The procedure also checks $\beta(e) = b$. Since the new action is regular, β is therefore the unique permutation in P_K corresponding to the element b . In other words, β is left translation by b in the group $(K, \circ_K)$.

Finally,

$$P_K = \langle \{\hat{\gamma} : \gamma \in \Gamma_H\} \cup \{\beta\} \rangle.$$

The permutations $\{\hat{\gamma} : \gamma \in \Gamma_H\}$ generate the embedded copy of P_H , which corresponds to the subgroup H , while β corresponds to the element b . Therefore the induced group on K is generated by H and b :

$$K = \langle H, b \rangle_{\circ_K}.$$

□

3.2 Testing whether a transitive action is regular

In this section, we give a fast test that never rejects a regular action and, with high probability rejects a non-regular one.

Theorem 3.8 (Randomized regularity test). *Let X have size k , let $e \in X$, and let*

$$\Gamma = \{\gamma_1, \dots, \gamma_m\} \subseteq \text{Sym}(X).$$

For every $0 < \delta < 1$, there is a randomized algorithm running in

$$O\left(km \log \frac{1}{\delta}\right)$$

time such that a regular action is always accepted, while a non-regular action is rejected with probability at least $1 - \delta$. A non-transitive action is rejected deterministically.

Proof. Let $P := \langle \Gamma \rangle \leq \text{Sym}(X)$. As above, every permutation in Γ is stored explicitly by its values on X , so evaluating $\gamma(x)$ takes constant time.

We first compute the inverses of all permutations in Γ , and set $\Sigma := \Gamma \cup \Gamma^{-1}$. Thus $|\Sigma| \leq 2m$ and $P = \langle \Sigma \rangle$. Computing the inverse of one explicitly stored permutation takes $O(k)$ time, so all inverses can be computed in $O(km)$ time.

We next compute the orbit of e ,

$$\text{Orb}_P(e) := \{q(e) : q \in P\}.$$

This can be done directly from the generators in Σ . Starting with e , whenever a point x is discovered, we compute $\gamma(x)$ for every $\gamma \in \Sigma$, and add every new point to the orbit. Since there are at most k points and at most $2m$ generators, this takes $O(km)$ time. If $\text{Orb}_P(e) \neq X$, then the action is not transitive, so the algorithm rejects deterministically.

Suppose from now on that the action is transitive. While computing the orbit, we also construct an *orbit spanning tree* rooted at e . Whenever a new point $z = \gamma(x)$ is first discovered from an already discovered point x , we add the directed tree edge $x \xrightarrow{\gamma} z$. Set $p_e := 1$, the identity permutation. For every $x \in X$, the unique path in the tree from e to x defines a permutation $p_x \in P$ satisfying $p_x(e) = x$. We do not explicitly construct the permutation p_x for every x ; instead, the path from e to x in the tree implicitly represents p_x .

For every $x \in X$ and every $\gamma \in \Sigma$, define

$$s_{x,\gamma} := p_{\gamma(x)}^{-1} \gamma p_x.$$

Then

$$s_{x,\gamma}(e) = p_{\gamma(x)}^{-1} \gamma p_x(e) = p_{\gamma(x)}^{-1}(\gamma(x)) = e.$$

Hence $s_{x,\gamma} \in P_e$, where $P_e := \{q \in P : q(e) = e\}$ is the stabilizer of e .

We claim that the permutations $\{s_{x,\gamma} : x \in X, \gamma \in \Sigma\}$ generate P_e . Indeed, let $q = \gamma_r \cdots \gamma_1 \in P_e$, $\gamma_i \in \Sigma$, and define

$$x_0 := e, \quad x_i := \gamma_i \cdots \gamma_1(e) \quad (1 \leq i \leq r).$$

Since $q \in P_e$, $x_r = q(e) = e$. Therefore

$$\begin{aligned} s_{x_{r-1},\gamma_r} s_{x_{r-2},\gamma_{r-1}} \cdots s_{x_0,\gamma_1} &= p_{x_r}^{-1} \gamma_r p_{x_{r-1}} p_{x_{r-1}}^{-1} \gamma_{r-1} p_{x_{r-2}} \cdots p_{x_1}^{-1} \gamma_1 p_{x_0} \\ &= p_e^{-1} \gamma_r \cdots \gamma_1 p_e = q. \end{aligned}$$

Thus the permutations $s_{x,\gamma} = p_{\gamma(x)}^{-1} \gamma p_x$ generate the stabilizer P_e .

We now describe one randomized trial. Choose a uniformly random point $y \in X$. For every $x \in X$, define $v_y(x) := p_x(y)$. Although the permutations p_x are represented only implicitly by the orbit tree, all the values $v_y(x)$ can be computed in $O(k)$ time by a single traversal of that tree: Start with $v_y(e) := y$. Suppose the tree contains an edge $x \xrightarrow{\gamma} z$. By the definition of the tree permutation, $p_z = \gamma p_x$. Hence, once $v_y(x)$ has been computed,

$$v_y(z) = p_z(y) = \gamma p_x(y) = \gamma(v_y(x)).$$

Thus every tree edge is processed only once, and all k values $v_y(x)$ are computed in $O(k)$ time.

After computing these values, for every $x \in X$ and $\gamma \in \Sigma$, we check whether

$$\gamma(v_y(x)) = v_y(\gamma(x)). \quad (*)$$

Using the definition of v_y , condition $(*)$ is equivalent to $\gamma p_x(y) = p_{\gamma(x)}(y)$. Applying $p_{\gamma(x)}^{-1}$ to both sides, this is equivalent to $p_{\gamma(x)}^{-1} \gamma p_x(y) = y$, or $s_{x,\gamma}(y) = y$. Since the permutations $s_{x,\gamma}$

generate P_e , all the checks in the trial pass if and only if every element of P_e fixes y . Equivalently, the trial passes if and only if $y \in \text{Fix}(P_e)$, where

$$\text{Fix}(P_e) := \{y \in X : q(y) = y \text{ for every } q \in P_e\}.$$

If the action is regular, then $P_e = \{1\}$. Therefore every point $y \in X$ belongs to $\text{Fix}(P_e)$, so every trial passes. Hence a regular action is always accepted.

Now suppose that the action is transitive but not regular. Then $P_e \neq \{1\}$. We claim that

$$|\text{Fix}(P_e)| \leq \frac{k}{2}. \quad (**)$$

Let $F := \text{Fix}(P_e)$. For every $y \in X$, let $P_y := \{q \in P : q(y) = y\}$ be the stabilizer of y . If $y \in F$, then every element of P_e fixes y , and hence $P_e \subseteq P_y$. Since the action is transitive, all point stabilizers have the same size. Therefore $|P_e| = |P_y|$, and hence $P_e = P_y$. Conversely, if $P_y = P_e$, then every element of P_e fixes y , so $y \in F$. Thus

$$F = \{y \in X : P_y = P_e\}.$$

Because the action is transitive, every $y \in X$ can be written as $y = q(e)$ for some $q \in P$. For such a point, $P_{q(e)} = qP_eq^{-1}$. Consequently, $q(e) \in F$ if and only if $qP_eq^{-1} = P_e$. Therefore

$$F = \{q(e) : q \in N_P(P_e)\},$$

where $N_P(P_e) := \{q \in P : qP_eq^{-1} = P_e\}$ is the normalizer of P_e in P . So F is exactly the orbit of e under the group $N_P(P_e)$. Hence, by the orbit-stabilizer theorem,

$$|F| = [N_P(P_e) : P_e].$$

We next show that $N_P(P_e) \neq P$. Suppose otherwise that $N_P(P_e) = P$. Then P_e is normal¹⁰ in P . Since the action is transitive, every point $y \in X$ has the form $y = q(e)$, and therefore $P_y = qP_eq^{-1} = P_e$. Thus every element of P_e fixes every point of X . But $P \leq \text{Sym}(X)$, so the only permutation fixing every point of X is the identity. Hence $P_e = \{1\}$, contradicting the assumption that the action is not regular.

Thus $N_P(P_e)$ is a proper subgroup of P , so

$$[P : N_P(P_e)] \geq 2.$$

By the orbit-stabilizer theorem and transitivity, $k = [P : P_e]$. Therefore

$$|F| = [N_P(P_e) : P_e] = \frac{[P : P_e]}{[P : N_P(P_e)]} \leq \frac{k}{2}.$$

This proves (**).

It follows that, when the action is transitive but not regular, one random trial passes with probability at most $1/2$. Repeat the trial independently $r := \lceil \log(1/\delta) \rceil$ times, and accept only if every trial passes. A nonregular transitive action is then accepted with probability at most $2^{-r} \leq \delta$. Hence it is rejected with probability at least $1 - \delta$.

Finally, we bound the running time. Computing the inverse generators, the orbit of e , and the orbit spanning tree takes $O(km)$ time. In one randomized trial, all values $v_y(x) = p_x(y)$ are computed by one traversal of the tree in $O(k)$ time. We then perform the checks $\gamma(v_y(x)) = v_y(\gamma(x))$ for every $x \in X$ and every $\gamma \in \Sigma$. Since $|\Sigma| \leq 2m$, these checks take $O(km)$ time. Thus one trial takes $O(km)$ time. With $r = O(\log(1/\delta))$ trials, the total running time is

$$O\left(km \log \frac{1}{\delta}\right).$$

This proves the theorem. □

¹⁰For every $q \in P$ we have $qP_eq^{-1} = P_e$.

3.3 Reconstruction on all labels

We first identify an identity candidate. Choose any $a \in G$ and query both $a * x$ and $x * a$ for all $x \in G$. Let $E_a := \{x \in G : a * x = x * a = a\}$. Require that E_a consist of exactly one label, call it e , and then verify globally that $e * x = x * e = x$ for every $x \in G$. This uses $O(n)$ queries and is always correct on a genuine group.

Start with $H_0 = \{e\}$ and the trivial action. Put $L := \lceil \log_2 n \rceil$ and $\delta_{\text{reg}} := 1/(100L)$. While $H_i \neq G$, choose any $b_i \in G \setminus H_i$, run Explore-Extension, and then run the regularity test on the returned action with error δ_{reg} . Reject if either procedure rejects. Otherwise continue with the returned set H_{i+1} .

By the unconditional doubling lemma, every accepted extension doubles the current set, so at most L extensions occur. Let **BadReg** be the event that some nonregular action is mistakenly accepted. A union bound gives

$$\Pr[\text{BadReg}] \leq L\delta_{\text{reg}} \leq \frac{1}{100}. \quad (3.3.1)$$

Conditioned on $\neg \text{BadReg}$, the certified-extension lemma shows inductively that every accepted stage is a genuine group under the operation induced by its regular action. Hence, if reconstruction reaches all of G , it produces a genuine group operation $\circ : G \times G \rightarrow G$. On a genuine input group, the honest-extension lemma shows more: every stage is the true subgroup extension in $(G, *)$, every regularity test accepts with probability one, and the final action is the true left-regular action of $(G, *)$.

Query complexity of reconstruction. Let $1 = h_0 < h_1 < \dots < h_r = n$ be the sizes of accepted stages. Since $h_{i+1} \geq 2h_i$,

$$\sum_{i=0}^{r-1} (2h_{i+1} - h_i) = 2h_r + \sum_{i=1}^{r-1} h_i - h_0 < 3n.$$

Lemma 3.5 together with the identity queries and one possible final failed extension, reconstruction uses $O(n)$ queries on every input. At stage i there are only $O(i) = O(\log n)$ explicit permutation generators. By Theorem 3.8, constructing their blockwise extensions and running the regularity test on a stage of size k takes $O(k \log n \log \log n)$ time. The accepted stage sizes form a geometric sequence, so the whole reconstruction takes

$$O(k \log n \log \log n) \quad (3.3.2)$$

time.

3.4 Evaluating the reconstructed group quickly

A regular action gives a group operation, but naively finding the unique permutation corresponding to a label can be expensive. We now build a query-free data structure for fast multiplication.

Fix an orbit spanning tree of the final transitive action, rooted at e . For each $x \in G$, let p_x be the tree word with $p_x(e) = x$. For $s \in G$, define the syntactic column $T_s(x) := p_x(s)$. The entire column can be computed in $O(n)$ time by traversing the orbit tree. If the final action is regular and induces $\circ$, then, by Lemma 2.2, p_x is left multiplication by x , so

$$T_s(x) = x \circ s. \quad (3.4.1)$$

One preprocessing attempt does the following. Sample $O(\log n)$ uniform labels s . For each one, compute its full column T_s , find the unique label r satisfying $T_s(r) = e$, and compute T_r as

well. If the inverse label is not unique, declare failure. Use all sampled labels and their recorded inverses as generators of a directed graph with edges $x \rightarrow T_s(x)$. Run breadth-first search from e . Declare failure unless every label is reached within depth $O(\log n)$. On success, store one short word

$$W(y) = (s_1, \dots, s_\ell), \quad \ell = O(\log n),$$

for every y .

When the action is regular, (3.4.1) says these are the true right Cayley edges of $(G, \circ)$. Then Corollary 2.4 implies that one attempt succeeds with probability at least $2/3$. Moreover, whenever it succeeds, define $\text{Eval}(x, e) = x$ and, if $W(y) = (s_1, \dots, s_\ell)$,

$$\text{Eval}(x, y) = T_{s_\ell}(\dots T_{s_2}(T_{s_1}(x)) \dots).$$

Then $\text{Eval}(x, y) = x \circ y$ for every pair. One preprocessing attempt costs $O(n \log n)$ time, makes no queries to the original table, and a successful evaluation costs $O(\log n)$ time.

3.5 The complete tester

The tester is now simple.

1. Set $\bar{\epsilon} = \max\{\epsilon, n^{-2}\}$.
2. Find and verify the identity, and run the reconstruction above. Reject if reconstruction rejects.
3. Independently run the multiplication preprocessing three times, stopping at the first success. If all three attempts fail, accept.
4. If preprocessing succeeds, choose $q = \lceil 5/\bar{\epsilon} \rceil$ independent uniform pairs (x, y) . Query $x * y$ and compare it with $\text{Eval}(x, y)$. Reject at the first disagreement; accept if all comparisons agree.

The acceptance after three preprocessing failures is useful only to obtain perfect completeness. For the present two-sided theorem this extra feature is harmless.

Lemma 3.9 (Completeness). *If $(G, *)$ is a group, the tester accepts with probability 1.*

Proof. The identity step finds the true identity. By the honest-extension lemma, reconstruction returns the true left-regular action, and every regularity test accepts. If all preprocessing attempts fail, the tester explicitly accepts. Otherwise the evaluator computes the true product $x * y$ on every pair, so no comparison can reject. $\square$

Lemma 3.10 (Soundness). *If $*$ is ϵ -far from every group operation on G , the tester rejects with probability greater than $2/3$.*

Proof. By (3.3.1), the probability that a nonregular action is erroneously accepted during reconstruction is at most $1/100$. Condition on the complementary event. If reconstruction rejects, we are done. Otherwise the final regular action induces a genuine group operation $\circ$ on G .

Each preprocessing attempt succeeds with probability at least $2/3$, using fresh randomness. Hence the probability that all three fail is at most $(1/3)^3 = 1/27$. On every successful attempt, the evaluator computes $x \circ y$. Since $*$ is $\bar{\epsilon}$ -far from every group, a uniform pair disagrees with $\circ$ with probability at least $\bar{\epsilon}$. Therefore the chance that all $q = \lceil 5/\bar{\epsilon} \rceil$ random comparisons miss the disagreement set is at most

$$(1 - \bar{\epsilon})^q \leq e^{-5} < \frac{1}{100}.$$

Thus the total false-acceptance probability is at most $1/100 + 1/27 + 1/100 < 1/3$. $\square$

Proof of Theorem 3.1. Completeness and soundness are Lemmas 3.9 and 3.10. Reconstruction uses $O(n)$ queries, preprocessing uses none, and the final comparison uses $O(1/\bar{\epsilon}) = O(1/\epsilon)$ queries. Thus the total query complexity is

$$O\left(n + \frac{1}{\epsilon}\right).$$

By (3.3.2) the reconstruction time is $O(n \log n \log \log n)$. Three preprocessing attempts cost $O(n \log n)$, and the final $O(1/\bar{\epsilon})$ evaluations cost $O((\log n)/\bar{\epsilon})$. This gives

$$O\left(n \log n \log \log n + \frac{\log n}{\epsilon}\right).$$

At most $O(\log n)$ degree- n permutations and $O(\log n)$ columns are stored, giving $O(n \log n)$ words of space. $\square$

4 Dictionary and locator machinery

The sublinear tester will use a small random dictionary and a common random-walk locator. This section develops that machinery and two structural lemmas. We allow arbitrary finite postprocessing: in particular, we enumerate all abstract groups of order n and all possible interpretations of a small number of visible labels. The number of candidates may be enormous. However, the multiplication table is queried only once to produce the visible trajectories used by the test. Each candidate is then checked against these same already queried trajectories by computation alone. Thus enumerating many candidates may increase the running time, but it does not multiply the number of oracle queries.

4.1 Random generator labels and inverse candidates

Choose $k := \lceil c_{\text{AR}} \log(2n) \rceil$ independent uniform labels $s_1, \dots, s_k \in G$ where c_{AR} is the constant in Theorem 2.3. For each s_i , evaluate a fixed parenthesized expression for $t_i := s_i^{n-1}$ using a binary addition chain. This is syntactically meaningful even when $*$ is not associative, because the parenthesization is fixed in advance. It uses $O(\log n)$ queries per i , hence $O(\log^2 n)$ in total. On a genuine group, Lagrange's theorem gives $t_i = s_i^{-1}$. Write

$$u_{+i} := s_i, \quad u_{-i} := t_i.$$

The soundness proof never assumes that the t_i are true inverses.

4.2 A random dictionary

Let m be a parameter with $1 \leq m \leq c_D n$, where $c_D \leq 1/500$. Choose a uniform starting label $d_0 \in G$ and independent uniform signed indices $\xi_1, \dots, \xi_m \in \{\pm 1, \dots, \pm k\}$. Query the walk

$$d_j := d_{j-1} * u_{\xi_j}, \quad 1 \leq j \leq m.$$

Let $D := \{d_0, \dots, d_m\}$ be the set of distinct labels visited. Reject if $|D| < m/2$.

We denote by

$$\mathcal{E}_{\text{spec}} := \left\{ \left\| P - \frac{J}{n} \right\|_{2 \rightarrow 2} \leq \lambda_0 \right\} \quad (4.2.1)$$

the good spectral event. See Theorem 2.3.

Lemma 4.1 (The dictionary is large on a group). *Condition on the input being a genuine group and on the spectral event $\mathcal{E}_{\text{spec}}$ occurring. There is an absolute integer n_0 such that, for every $n \geq n_0$ and every $1 \leq m \leq c_D n$,*

$$\Pr[|D| \geq m/2] \geq \frac{99}{100}.$$

Proof. Because d_0 is uniform on G , every point d_i of the random walk is also uniform on G . Equivalently, the uniform distribution is *stationary* for this walk. Let

$$C_{\text{coll}} := |\{(r, s) : 0 \leq r < s \leq m, d_r = d_s\}|.$$

If $1 = \lambda_1, \lambda_2, \dots, \lambda_n$ are the eigenvalues of the normalized transition matrix P , then condition on the spectral event in Theorem 2.3 with the fixed value λ_0 and (A.3.1) in Appendix A.3, $|\lambda_i| \leq \lambda_0$ for all $i \geq 2$. If two walk positions are t steps apart, then

$$\Pr[d_r = d_{r+t}] = \frac{1}{n} \sum_{x \in G} P^t(x, x) = \frac{1}{n} \text{tr}(P^t) = \frac{1}{n} \sum_{i=1}^n \lambda_i^t \leq \frac{1}{n} (1 + (n-1)\lambda_0^t) \leq \frac{1}{n} + \lambda_0^t.$$

Therefore

$$\begin{aligned} \mathbb{E}[C_{\text{coll}}] &\leq \binom{m+1}{2} \frac{1}{n} + \sum_{t=1}^m (m+1-t) \lambda_0^t \\ &\leq \frac{(m+1)^2}{2n} + (m+1) \frac{\lambda_0}{1-\lambda_0}. \end{aligned}$$

Every repeated visit can reduce the number of distinct visited labels by at most one, so $m+1-|D| \leq C_{\text{coll}}$. Dividing the expectation bound by m and using $m \geq 1$ and $m \leq c_D n$ gives

$$\begin{aligned} \frac{\mathbb{E}[C_{\text{coll}}]}{m} &\leq \frac{(m+1)^2}{2nm} + \frac{m+1}{m} \frac{\lambda_0}{1-\lambda_0} \\ &\leq \frac{c_D}{2} + \frac{3}{2n} + \frac{2\lambda_0}{1-\lambda_0}. \end{aligned}$$

Choose an absolute integer $n_0 \geq 1500$. Together with the fixed choices of λ_0 and c_D above, this implies, for every $n \geq n_0$, $\mathbb{E}[C_{\text{coll}}]/m \leq 3/1000 < 1/200$. If $|D| < m/2$, then $C_{\text{coll}} > m/2$. Markov's inequality therefore gives

$$\Pr[|D| < m/2] \leq \Pr[C_{\text{coll}} > m/2] \leq \frac{2\mathbb{E}[C_{\text{coll}}]}{m} \leq \frac{6}{1000} < \frac{1}{100}.$$

This proves the claim. $\square$

Fix once and for all such an absolute integer n_0 . Whenever $n < n_0$, the sublinear testers below use the linear-query tester instead.

4.3 Enumerating abstract candidate groups

Computational efficiency is now ignored. Enumerate one representative H of every isomorphism class of groups of order n . For each H , enumerate all marked tuples $(\rho, a_1, \dots, a_k) \in H^{k+1}$. Let $g(n)$ be the number of isomorphism classes. The total number of marked candidates is at most $M := g(n)n^{k+1}$. Let $\Lambda := \log(3M)$. By Pyber's Theorem 2.5, $g(n) \leq \exp(C(\log n)^3)$ and since $k = O(\log n)$,

$$\Lambda = O(\log^3 n).$$

For a candidate $\mathcal{H} = (H, \rho, a_1, \dots, a_k)$, set $a_{-i} = a_i^{-1}$ and imitate the dictionary word inside H : $h_0 := \rho$, and $h_j := h_{j-1}a_{\xi_j}$. Retain the candidate only when the equality patterns agree exactly:

$$d_r = d_s \iff h_r = h_s \quad (0 \leq r, s \leq m). \quad (4.3.1)$$

For every retained candidate, this defines an injective map

$$\phi_{\mathcal{H}} : D \rightarrow H, \quad \phi_{\mathcal{H}}(d_j) = h_j.$$

If no candidate remains, reject.

4.4 The common locator

Choose independently one common random *locator word* $\sigma_1, \dots, \sigma_T \in \{\pm 1, \dots, \pm k\}$. For a starting label $x \in [n]$, define $x_0 = x$, $x_j = x_{j-1} * u_{\sigma_j}$. When the locator is evaluated at x , follow the path $x = x_0, x_1, \dots$ and stop at the first index t for which $x_t \in D$. If no such index occurs within T steps, stop after T steps. Let

$$\tau(x) := \min\{0 \leq j \leq T : x_j \in D\},$$

with $\tau(x) = \infty$ if there is no hit.

For a retained candidate (satisfies (4.3.1)), define

$$g_0(\mathcal{H}) = 1_H, \quad g_j(\mathcal{H}) = a_{\sigma_1} \cdots a_{\sigma_j},$$

and define a partial map $\Psi_{\mathcal{H}} : G \dashrightarrow H$ by

$$\Psi_{\mathcal{H}}(x) := \phi_{\mathcal{H}}(x_{\tau(x)}) g_{\tau(x)}(\mathcal{H})^{-1} \quad (4.4.1)$$

when $\tau(x) < \infty$. If the path misses D , leave $\Psi_{\mathcal{H}}(x)$ undefined. We call $\Psi_{\mathcal{H}}$ the *locator map*.

The same visible locator path is used for all candidates. Therefore one path is queried once and interpreted offline in every candidate group; the enormous number M of candidates does not multiply the oracle-query cost.

We next show that in a genuine group a suitable locator misses only a small fraction of starting labels.

Lemma 4.2 (Killed-walk estimate). *Let P be a normalized transition matrix satisfying*

$$\left\| P - \frac{J}{n} \right\|_{2 \rightarrow 2} \leq \lambda < 1.$$

Let $D \subset G$ have density $\alpha = |D|/n$, let $B = G \setminus D$, and put $Q := \Pi_B P \Pi_B$, where Π_B is the diagonal projection onto coordinates in B .¹¹ Then

$$\|Q\|_{2 \rightarrow 2} \leq 1 - (1 - \lambda)\alpha. \quad (4.4.2)$$

Proof. Let $\gamma = 1 - (1 - \lambda)\alpha$. Let $f : G \rightarrow \mathbb{R}$ be supported on B , i.e., it satisfy $f(x) = 0$ for every $x \notin B$. Write $f = a\mathbf{1} + f_0$, where $f_0 \perp \mathbf{1}$. Since $f_0 \perp \mathbf{1}$, we have $(J/n)f_0 = 0$. Hence

$$\|Pf_0\|_2 = \left\| \left(P - \frac{J}{n} \right) f_0 \right\|_2 \leq \lambda \|f_0\|_2.$$

By Cauchy–Schwarz, $|\langle f_0, Pf_0 \rangle| \leq \|f_0\|_2 \|Pf_0\|_2 \leq \lambda \|f_0\|_2^2$, which gives

$$-\lambda \|f_0\|_2^2 \leq \langle f_0, Pf_0 \rangle \leq \lambda \|f_0\|_2^2.$$

Now, since $\|f\|_2^2 = \|f_0\|_2^2 + a^2 n$,

$$\langle f, Pf \rangle = a^2 n + \langle f_0, Pf_0 \rangle \leq a^2 n + \lambda \|f_0\|_2^2 = \lambda \|f\|_2^2 + (1 - \lambda)a^2 n.$$

¹¹ $(\Pi_B)_{x,y} = 1$ if $x = y \in B$, and $= 0$ otherwise. So $(\Pi_B P \Pi_B)_{x,y} = P_{x,y}$ if $x, y \in B$, and $= 0$ otherwise.

Because f is supported on $(1 - \alpha)n$ coordinates, Cauchy–Schwarz gives

$$a^2 n = \frac{\langle f, \mathbf{1} \rangle^2}{n} \leq \frac{\|f\|_2^2 \|\mathbf{1}_B\|_2^2}{n} = (1 - \alpha) \|f\|_2^2.$$

Thus

$$\langle f, Pf \rangle \leq (1 - (1 - \lambda)\alpha) \|f\|_2^2 = \gamma \|f\|_2^2.$$

The lower bound is

$$\langle f, Pf \rangle = a^2 n + \langle f_0, Pf_0 \rangle \geq -\lambda \|f\|_2^2 \geq -(1 - (1 - \lambda)\alpha) \|f\|_2^2 = -\gamma \|f\|_2^2.$$

It remains to translate this bound to Q . Let μ be a nonzero eigenvalue of Q , and let v be a corresponding eigenvector. Since the range of $Q = \Pi_B P \Pi_B$ is supported on B , the identity $Qv = \mu v$ implies that v is supported on B . Therefore,

$$|\mu| \|v\|_2^2 = |\langle v, Qv \rangle| = |\langle v, Pv \rangle| \leq \gamma \|v\|_2^2.$$

Thus $|\mu| \leq \gamma$. The same conclusion is immediate for the zero eigenvalues. Since Q is symmetric, its operator norm is the largest absolute value of its eigenvalues, and hence

$$\|Q\|_{2 \rightarrow 2} \leq \gamma = 1 - (1 - \lambda)\alpha.$$

□

Set

$$T := \left\lceil C_L \frac{n}{m} \log \frac{1}{\epsilon} \right\rceil \quad (4.4.3)$$

for a sufficiently large absolute constant C_L . For one fixed common locator word define the missed fraction

$$Z := \frac{1}{n} |\{x \in G : \tau(x) = \infty\}|. \quad (4.4.4)$$

Lemma 4.3 (Small missed fraction). *Suppose that the input is a genuine group, that $\mathcal{E}_{\text{spec}}$ occurs, and that $|D| \geq m/2$. For T as in (4.4.3),*

$$\Pr \left[Z > \frac{\bar{\epsilon}}{1000} \right] \leq \frac{1}{100}.$$

Proof. Condition on the Cayley graph and on D . Although all starting points use the same locator word and their hitting events are highly dependent, the marginal trajectory from any fixed starting point is an ordinary random walk with transition matrix P . Therefore, using only linearity of expectation,

$$\begin{aligned} \mathbb{E}[Z] &= \frac{1}{n} \mathbf{1}_B^T Q^T \mathbf{1}_B = \frac{1}{n} \langle \mathbf{1}_B, Q^T \mathbf{1}_B \rangle \leq \frac{1}{n} \|\mathbf{1}_B\|_2 \|Q^T \mathbf{1}_B\|_2 \leq \frac{1}{n} \|Q^T\|_{2 \rightarrow 2} \|\mathbf{1}_B\|_2^2 \\ &\leq \frac{|B|}{n} \|Q\|_{2 \rightarrow 2}^T \leq (1 - (1 - \lambda)\alpha)^T \leq \exp(-(1 - \lambda_0)\alpha T), \end{aligned}$$

where $\alpha = |D|/n \geq m/(2n)$. Choosing C_L large enough makes this expectation at most $\bar{\epsilon}/100000$. Markov's inequality then gives the stated probability. No independence between different starting points is used. □

Lemma 4.4 (The true candidate is located correctly). *Suppose the input is a genuine group and $\theta : (G, *) \rightarrow H$ is an isomorphism onto one of the enumerated representatives. Mark the candidate by $\rho = \theta(d_0)$, $a_i = \theta(s_i)$. Then this candidate is retained, and wherever its locator is defined,*

$$\Psi_{\mathcal{H}}(x) = \theta(x).$$

Proof. On a group, $t_i = s_i^{-1}$. Induction along the dictionary walk gives $h_j = \theta(d_j)$. Since θ is bijective, the equality pattern (4.3.1) holds exactly, so the candidate is retained and $\phi_{\mathcal{H}}(d) = \theta(d)$ on D . If $r = \tau(x) < \infty$, then the locator path uses right multiplication, so

$$\theta(x_r) = \theta(x)a_{\sigma_1} \cdots a_{\sigma_r} = \theta(x)g_r(\mathcal{H}).$$

Hence

$$\Psi_{\mathcal{H}}(x) = \phi_{\mathcal{H}}(x_r)g_r(\mathcal{H})^{-1} = \theta(x_r)g_r(\mathcal{H})^{-1} = \theta(x).$$

□

4.5 Defect and distance from a bijection

For a partial map $\Psi : G \dashrightarrow H$ with $|G| = |H| = n$, define

$$\text{defect}(\Psi) := 1 - \frac{|\text{Im } \Psi|}{n}.$$

Lemma 4.5 (Defect identity). *For every partial map $\Psi : G \dashrightarrow H$,*

$$\text{defect}(\Psi) = \min_{\pi: G \rightarrow H \text{ bijective}} \Pr_{x \in G}[\Psi(x) \text{ undefined or } \Psi(x) \neq \pi(x)].$$

Proof. Let $r = |\text{Im } \Psi|$. A bijection can agree with Ψ on at most one point from each nonempty fiber $\Psi^{-1}(h)$, hence on at most r points. Conversely, choose one representative from every nonempty fiber, prescribe the bijection there, and extend this partial injection arbitrarily to a bijection. The maximum possible number of agreements is exactly r , which proves the formula. □

4.6 Near-bijection plus approximate multiplication

For a partial map $\Psi : G \dashrightarrow H$, define its multiplicativity failure probability by

$$\eta(\Psi) := \Pr_{x, y \in G}[\Psi(x), \Psi(y), \Psi(x * y) \text{ are not all defined or } \Psi(x * y) \neq \Psi(x)\Psi(y)].$$

Lemma 4.6 (Near-bijection plus multiplicativity). *If H is a group of order n and $\Psi : G \dashrightarrow H$ satisfies $\text{defect}(\Psi) \leq \delta$, then there is a genuine group operation $\circ$ on G such that*

$$\text{dist}(*, \circ) \leq \eta(\Psi) + 3\delta.$$

Proof. By Lemma 4.5, choose a bijection $\pi : G \rightarrow H$ and a set $B \subseteq G$, $|B| \leq \delta n$, such that $\Psi(x) = \pi(x)$ ($x \notin B$). Transport the multiplication of H to G : $x \circ y := \pi^{-1}(\pi(x)\pi(y))$. This is a genuine group operation.

Consider a pair (x, y) for which $\Psi(x)$, $\Psi(y)$, and $\Psi(x * y)$ are all defined and $\Psi(x * y) = \Psi(x)\Psi(y)$. If $x, y, x * y \notin B$, then $\pi(x * y) = \Psi(x * y) = \Psi(x)\Psi(y) = \pi(x)\pi(y)$, so $x * y = x \circ y$.

Thus, if $\Psi(x * y) = \Psi(x)\Psi(y)$ holds, the only remaining case in which $x * y$ may differ from $x \circ y$ is when $x, y \notin B$ but $x * y \in B$. Define $\Psi(B) := \{\Psi(b) : b \in B \cap \text{Dom } \Psi\}$. Then

$$\pi(x)\pi(y) = \Psi(x)\Psi(y) = \Psi(x * y) \in \Psi(B).$$

For independent uniform x, y , the elements $\pi(x), \pi(y)$ are independent uniform elements of H , so their product is uniform. Therefore

$$\Pr[x, y \notin B \text{ and } x * y \in B] \leq \Pr[\pi(x)\pi(y) \in \Psi(B)] \leq \frac{|\Psi(B)|}{n} \leq \delta.$$

A disagreement $x * y \neq x \circ y$ is thus contained in the union of the event that $\Psi(x), \Psi(y), \Psi(x * y)$ are not all defined, $\Psi(x * y) \neq \Psi(x)\Psi(y)$, $x \in B$, $y \in B$, and the event: $x, y \notin B$ and $x * y \in B$. Their probabilities sum to at most $\eta(\Psi) + 3\delta$. □

5 A sublinear adaptive tester using action consistency

We now combine the dictionary and locator from Section 4 with a simple local consistency test. The main idea is that, when the candidate Cayley graph is an expander, a partial map that is almost consistent with the group action cannot map too many input elements to the same group element. Therefore, its image must contain almost all elements of the candidate group. This allows us to avoid reconstructing the whole group and gives a $\tilde{O}(\sqrt{n/\epsilon} + 1/\epsilon)$ query bound.

Theorem 5.1 ($\sqrt{n/\epsilon}$ -type adaptive upper bound). *For every $n \geq 2$ and $0 < \epsilon < 1/2$, there is a randomized adaptive two-sided tester with query complexity*

$$O\left(\sqrt{\frac{n}{\epsilon}} (\log^{3/2} n) \sqrt{\log \frac{1}{\epsilon}} + \frac{\log^3 n}{\epsilon}\right) = \tilde{O}\left(\sqrt{\frac{n}{\epsilon}} + \frac{1}{\epsilon}\right).$$

In particular, for every fixed constant $\epsilon > 0$, the query complexity is $\tilde{O}(\sqrt{n})$.

5.1 Candidates with an expanding Cayley graph

We use exactly the same random visible generator labels $s_1, \dots, s_k$, inverse candidates $t_i = s_i^{n-1}$, signed visible labels $u_{\pm i}$, and abstract candidates $\mathcal{H} = (H, \rho, a_1, \dots, a_k)$ as in Section 4. Thus $k = \Theta(\log n)$, $M \leq g(n)n^{k+1}$, and $\Lambda := \log(3M) = O((\log(2n))^3)$. Let $\mathcal{J} := \{\pm 1, \dots, \pm k\}$, and $a_{-i} := a_i^{-1}$. For a candidate, define the *directed boundary* of $S \subseteq H$ by

$$\partial_{\mathcal{H}} S := \{(h, j) \in S \times \mathcal{J} : ha_j \notin S\}$$

and its *edge conductance* by

$$\Phi(\mathcal{H}) := \min_{0 < |S| \leq n/2} \frac{|\partial_{\mathcal{H}} S|}{|\mathcal{J}| |S|}. \quad (5.1.1)$$

Since computation is unrestricted, this number can be computed offline for every candidate.

Let $0 < \lambda_0 \leq 1/3000$ be the constant in the random-Cayley expansion black box Theorem 2.3, and fix

$$\Phi_0 := \frac{(1 - \lambda_0)}{4} > 0.$$

Then Φ_0 is a constant $0.2499 < \Phi_0 < 0.25$. The Cayley graph is $|\Sigma|$ -regular, since from every vertex there is one edge for each generator in the symmetric multiset Σ . Moreover, on the event $\|P - J/n\|_{2 \rightarrow 2} \leq \lambda_0$, every nontrivial eigenvalue of the normalized transition matrix P has absolute value at most λ_0 . Hence $\lambda_2(P) \leq \lambda_0$. By Cheeger's inequality for regular graphs [11], the edge conductance satisfies

$$\Phi = \min_{\substack{S \subseteq V \\ 0 < |S| \leq n/2}} \frac{|E(S, V \setminus S)|}{|\Sigma| |S|} \geq \frac{1 - \lambda_2(P)}{2} \geq \frac{1 - \lambda_0}{2}.$$

Therefore, the genuine candidate passes the filter

$$\Phi(\mathcal{H}) \geq \Phi_0 \quad (5.1.2)$$

whenever the spectral event holds. We discard every candidate that fails this filter.¹²

Choose a sufficiently small absolute constant $\gamma > 0$ so that

$$C_{\Phi} \gamma \leq \frac{1}{160}, \quad \gamma \leq \frac{1}{400} \quad \text{where } C_{\Phi} := \frac{(1 + \Phi_0)}{\Phi_0}. \quad (5.1.3)$$

¹²Although H is a group, the filter depends on the chosen marks $a_1, \dots, a_k$. For the genuine candidate, where $a_i = \theta(s_i)$, the spectral event guarantees that the corresponding Cayley graph has conductance at least Φ_0 . Other choices of the a_i may give a Cayley graph with much smaller conductance, or even a disconnected one, and hence may fail the filter.

5.2 Dictionary and locator

Set

$$m = \Theta\left(\sqrt{\frac{n\Lambda \log(1/\bar{\epsilon})}{\bar{\epsilon}}}\right).$$

If $n < n_0$ or $m > c_D n$, we use the linear-query tester of Theorem 3.1. Otherwise $1 \leq m \leq c_D n$, and we proceed with this choice of m .

Construct the same dictionary walk as before: $d_r = d_{r-1} * u_{\xi_r}$, $1 \leq r \leq m$, and let $D = \{d_0, \dots, d_m\}$. Reject if $|D| < m/2$. By Lemma 4.1, on a genuine group this happens with probability at most $1/100$, conditioned on the spectral event.

For each candidate $\mathcal{H}$ imitate the dictionary word inside H : $h_0 = \rho$, $h_r = h_{r-1} a_{\xi_r}$. Retain the candidate only if it passes both the conductance filter (5.1.2) and the exact equality-pattern test

$$d_r = d_s \iff h_r = h_s \quad (0 \leq r, s \leq m).$$

Thus every retained candidate gives an injective dictionary map $\phi_{\mathcal{H}} : D \rightarrow H$ where $\phi_{\mathcal{H}}(d_r) = h_r$.

Choose an independent common locator word $\sigma_1, \dots, \sigma_L \in \mathcal{J}$ of length

$$L := \left\lceil C_L \frac{n}{m} \log \frac{1}{\bar{\epsilon}} \right\rceil.$$

Starting from x , follow $x_0 = x$, and $x_r = x_{r-1} * u_{\sigma_r}$ until the first hit on D , and define the same partial map as in (4.4.1):

$$\Psi_{\mathcal{H}}(x) = \phi_{\mathcal{H}}(x_{\tau(x)})(a_{\sigma_1} \cdots a_{\sigma_{\tau(x)}})^{-1}.$$

The proof of Lemma 4.3 works with any fixed small constant in place of $1/1000$. Increasing C_L if necessary gives the following form.

Lemma 5.2 (The fast locator misses few points). *On a genuine group, conditioned on the spectral event and $|D| \geq m/2$,*

$$\Pr[Z > \gamma \bar{\epsilon}] \leq \frac{1}{100},$$

where Z is the fraction of starting labels on which the common locator is undefined.

Proof. The killed-walk proof of Lemma 4.3 gives $\mathbb{E}[Z] \leq \exp(-c(m/n)L)$. With $L = C_L(n/m) \log(1/\bar{\epsilon})$, choosing C_L large enough makes the right side at most $\gamma \bar{\epsilon}/100$. Markov's inequality gives the claim. $\square$

For a genuine input, the true candidate is still interpreted correctly wherever the locator is defined, by Lemma 4.4. The conductance filter (5.1.2) does not remove it on the spectral event.

5.3 Action consistency forces near-bijection

Fix a retained candidate $\mathcal{H}$, so $\Phi(\mathcal{H}) \geq \Phi_0$, and a partial map $\Psi : G \dashrightarrow H$. An *action trial* chooses a uniform $x \in G$ and a uniform signed index $j \in \mathcal{J}$, queries $y := x * u_j$, and $z := y * u_{-j}$, and succeeds exactly when

$$\Psi(x), \Psi(y) \text{ are defined,} \quad \Psi(y) = \Psi(x) a_j, \quad z = x. \quad (5.3.1)$$

Let $\eta_{\text{act}}(\Psi)$ be its failure probability. Recall that $\text{defect}(\Psi) = 1 - |\text{Im } \Psi|/n$.

Lemma 5.3 (Action-defect lemma). *If $\Phi(\mathcal{H}) \geq \Phi > 0$, then*

$$\text{defect}(\Psi) \leq \frac{1 + \Phi}{\Phi} \eta_{\text{act}}(\Psi).$$

Proof. For $h \in H$, let $F_h := \{x \in G : \Psi(x) = h\}$, and $c_h := |F_h|$. Let

$$u := |\{x \in [n] : \Psi(x) \text{ is undefined}\}|$$

be the number of points on which Ψ is undefined. So

$$\sum_{h \in H} c_h = n - u.$$

For $h \in H$ and $j \in \mathcal{J}$, let $b_{h,j}$ be the number of $x \in F_h$ for which the action trial (x, j) fails.

If for $x \in F_h$ and $j \in \mathcal{J}$, the local action-consistency check succeeds, that is, if $\Psi(x * u_j) = \Psi(x)a_j = ha_j$, then $x * u_j \in F_{ha_j}$. This map $x \rightarrow x * u_j$ is injective on the successful points¹³: if two successful points x_1 and x_2 have the same image y , then the last condition in (5.3.1) gives $x_1 = y * u_{-j} = x_2$. Therefore $c_h - b_{h,j} \leq c_{ha_j}$, and $(c_h - c_{ha_j})_+ \leq b_{h,j}$. Applying the same argument with the inverse signed generator gives $(c_{ha_j} - c_h)_+ \leq b_{ha_j, -j}$. Hence

$$|c_h - c_{ha_j}| \leq b_{h,j} + b_{ha_j, -j}.$$

Summing over all h, j , and since H is a group,

$$V := \sum_{h,j} |c_h - c_{ha_j}| \leq 2 \sum_{h,j} b_{h,j} \leq 2\eta_{\text{act}}(\Psi)n|\mathcal{J}|. \quad (5.3.2)$$

Also, if $\Psi(x)$ is undefined, then the local action-consistency check fails for every generator u_j and every inverse generator u_{-j} . Hence

$$u \leq \eta_{\text{act}}(\Psi)n. \quad (5.3.3)$$

Let $r := |\{h : c_h > 0\}| = |\text{Im } \Psi|$ and for integer $t \geq 0$ define level sets $S_t := \{h : c_h > t\}$. For each $(h, j) \in H \times \mathcal{J}$, the integer $|c_h - c_{ha_j}|$ counts exactly the number of levels t for which one endpoint $v \in \{h, ha_j\}$ lies in S_t and the other does not. Since $\mathcal{J}$ is symmetric, every pair with one of h, ha_j in S_t and the other outside S_t appears twice in the sum over $H \times \mathcal{J}$: once as (h, j) and once as $(ha_j, -j)$. Only the one whose first element lies in S_t is counted in $\partial_{\mathcal{H}} S_t$. Therefore,

$$V = \sum_{h,j} |c_h - c_{ha_j}| = 2 \sum_{t \geq 0} |\partial_{\mathcal{H}} S_t|. \quad (5.3.4)$$

By conductance (5.1.1),

$$|\partial_{\mathcal{H}} S_t| \geq \Phi|\mathcal{J}| \min\{|S_t|, n - |S_t|\}. \quad (5.3.5)$$

Suppose first that $r \geq n/2$. Since S_0 is the image of Ψ , $|S_0| = |\text{Im } \Psi| = r \geq n/2$, and therefore

$$\min\{|S_0|, n - |S_0|\} = n - r. \quad (5.3.6)$$

For $t \geq 1$, every element of S_t has fiber size at least two, so $|S_t| \leq (n - u)/2 \leq n/2$. Moreover

$$\sum_{t \geq 1} |S_t| = \sum_h (c_h - 1)_+ = \sum_{h: c_h > 0} c_h - \sum_{h: c_h > 0} 1 = (n - u) - r. \quad (5.3.7)$$

Therefore, by (5.3.4), (5.3.5), (5.3.6) and (5.3.7)

$$\frac{V}{2} \geq \Phi|\mathcal{J}| \left[(n - r) + \sum_{t \geq 1} |S_t| \right] = \Phi|\mathcal{J}|(2(n - r) - u).$$

¹³We do not need the action trial starting from $x * u_j$ to be successful.

With (5.3.2) and (5.3.3) we get

$$\text{defect}(\Psi) = \frac{n-r}{n} \leq \frac{1}{2} \left(\frac{1+\Phi}{\Phi} \right) \eta_{\text{act}}(\Psi). \quad (5.3.8)$$

Now suppose $r < n/2$. Every level set has size below $n/2$, so, by (5.3.4) and (5.3.5) we have

$$\frac{V}{2} \geq \Phi |\mathcal{J}| \sum_{t \geq 0} |S_t| = \Phi |\mathcal{J}| (n - u).$$

Together with (5.3.2) and (5.3.3), $\Phi(1 - \eta_{\text{act}}(\Psi)) \leq \eta_{\text{act}}(\Psi)$, so $\eta_{\text{act}}(\Psi) \geq \Phi/(1 + \Phi)$. Since defect is at most one,

$$\text{defect}(\Psi) \leq 1 \leq \frac{1 + \Phi}{\Phi} \eta_{\text{act}}(\Psi).$$

This also dominates (5.3.8), proving the lemma. $\square$

5.4 The action and product tests

For each marked candidate $\mathcal{H}$ satisfying $\Phi(\mathcal{H}) \geq \Phi_0$, let $\Psi : [n] \dashrightarrow H$ be the associated partial map. For sufficiently large constant C_A , perform

$$R_{\text{act}} := \left\lceil C_A \frac{\Lambda}{\bar{\epsilon}} \right\rceil$$

independent action-consistency trials. Retain the candidate pair (H, Ψ) only if at most

$$4\gamma\bar{\epsilon}R_{\text{act}} \quad (5.4.1)$$

of these trials fail.

For a fixed candidate with $\eta_{\text{act}}(\Psi_{\mathcal{H}}) \geq 8\gamma\bar{\epsilon}$, a Chernoff bound gives passing probability at most $e^{-cC_A\gamma\bar{\epsilon}R_{\text{act}}} = M^{-c'C_A}$. By (5.1.3) and Lemma 5.3, choosing C_A sufficiently large and union-bounding over all M candidates shows that, except with probability $1/100$, every surviving candidate satisfies

$$\text{defect}(\Psi_{\mathcal{H}}) \leq C_{\Phi}\eta_{\text{act}}(\Psi_{\mathcal{H}}) < 8C_{\Phi}\gamma\bar{\epsilon} \leq \frac{\bar{\epsilon}}{20}. \quad (5.4.2)$$

For the true candidate, if both $\Psi(x)$ and $\Psi(x * u_j)$ are defined, then by Lemma 4.4, $\Psi(x) = \theta(x)$ and $\Psi(x * u_j) = \theta(x * u_j)$. Since $a_j = \theta(u_j)$, it follows that $\Psi(x * u_j) = \Psi(x)a_j$. Moreover, on a genuine input group, $u_{-j} = u_j^{-1}$, and hence $(x * u_j) * u_{-j} = x$. Therefore, for the true candidate, the action trial succeeds whenever both $\Psi(x)$ and $\Psi(x * u_j)$ are defined.

$$\eta_{\text{act}}(\Psi_{\mathcal{H}}) \leq \Pr[\Psi(x) \text{ is undefined}] + \Pr[\Psi(x * u_j) \text{ is undefined}].$$

Since x is uniform in the genuine group and right multiplication by u_j is a permutation, $x * u_j$ is also uniform. Therefore, each of the two probabilities is equal to Z , and hence, by the definition of Z in (4.4.4) and Lemma 5.2, with probability at least $99/100$, $\eta_{\text{act}}(\Psi) \leq 2Z \leq 2\gamma\bar{\epsilon}$. A Chernoff bound therefore shows that the true candidate passes (5.4.1) with overwhelming constant probability.

After the action test, we run a multiplicativity test using Lemma 4.6. Namely choose

$$R_{\text{prod}} := \left\lceil C_P \frac{\Lambda}{\bar{\epsilon}} \right\rceil$$

fresh uniform pairs (x, y) , query $z = x * y$, and call a trial a failure if a locator value is undefined or

$$\Psi(z) \neq \Psi(x)\Psi(y).$$

Retain a candidate only when at most $\bar{\epsilon}R_{\text{prod}}/4$ trials fail.

The structural lemma Lemma 4.6 says the following. If a candidate has defect at most δ and multiplicativity failure rate η , then it induces a genuine group operation at distance at most $\eta + 3\delta$ from the input. Hence on a $\bar{\epsilon}$ -far input, every candidate satisfying (5.4.2) must have

$$\eta \geq \bar{\epsilon} - 3\frac{\bar{\epsilon}}{20} = \frac{17}{20}\bar{\epsilon}. \quad (5.4.3)$$

A Chernoff bound and a union bound over all M candidates imply that, for sufficiently large C_P , the probability that any such candidate survives is at most $1/100$.

For the true candidate on a genuine group, by Lemma 5.2, with probability at least $99/100$, $Z \leq \gamma\bar{\epsilon}$. If $Z \leq \gamma\bar{\epsilon}$, each of $x, y, x * y$ is marginally uniform, and multiplicativity, $\Psi(x * y) = \Psi(x)\Psi(y)$, holds whenever all three locator values are defined. Thus, by (5.1.3), the multiplicity failure rate is

$$\eta \leq 3Z \leq 3\gamma\bar{\epsilon} < \frac{\bar{\epsilon}}{100}.$$

Another Chernoff bound shows that it passes the product test with high constant probability.

5.5 Completeness, soundness, and query complexity

The complete tester performs the generator stage in Section 4.1, dictionary stage in Section 4.2, candidate equality and conductance filters in Section 4.3, one common locator in Section 4.4, the action test, and finally the product test in Section 5.4. It accepts iff at least one candidate $\mathcal{H}$ survives.

Lemma 5.4 (Completeness). *A genuine group is accepted with probability at least $2/3$.*

Proof. In the branch where we use the tester of Theorem 3.1, the claim follows immediately. Otherwise, the random Cayley spectral event and the dictionary-size event fail with probability at most $2/100$. On their complement, the true marked candidate passes the conductance and equality-pattern filters. By Lemma 5.2, the event $Z \leq \gamma\bar{\epsilon}$ fails with probability at most $1/100$. Given these setup events, (5.3.1) and the true-candidate multiplicativity bound above show, via independent Chernoff bounds, that the action and product stages reject the true candidate with probability $e^{-\Omega(\Lambda)}$. Choosing the absolute constants large enough makes the total failure probability less than $1/3$. $\square$

Lemma 5.5 (Soundness). *If $*$ is ϵ -far from every group operation, the tester rejects with probability at least $2/3$.*

Proof. In the branch where we use the tester of Theorem 3.1, the claim follows immediately. Otherwise, condition on all setup randomness through the common locator. Each retained candidate is now a fixed partial map. The action samples are fresh, so except with probability at most $1/100$, every candidate that reaches the product stage has defect less than $\bar{\epsilon}/20$. For each such candidate (5.4.3) holds. The product samples are fresh, and the product-test union bound shows that, except with probability at most $1/100$, no low-defect candidate survives. Thus the total false-acceptance probability is below $1/3$. $\square$

It remains to count queries. Computing the inverse candidates costs $O(\log^2 n)$, and the dictionary costs m . By (4.4.3), every locator evaluation costs $O((n/m) \log(1/\bar{\epsilon}))$. The action and product stages together use $O(\Lambda/\bar{\epsilon})$ trials, each with only constantly many direct queries and locator evaluations. Therefore

$$Q(m) = O\left(\log^2 n + m + \frac{n \log(1/\bar{\epsilon})\Lambda}{m\bar{\epsilon}} + \frac{\Lambda}{\bar{\epsilon}}\right). \quad (8.23)$$

Balancing the second and third terms gives exactly $m = \sqrt{n \log(1/\bar{\epsilon}) \Lambda / \bar{\epsilon}}$. Thus, using $\Lambda = O((\log(2n))^3)$, we get

$$Q = O\left(\sqrt{\frac{n}{\bar{\epsilon}}}(\log(2n))^{3/2}\sqrt{\log \frac{1}{\bar{\epsilon}}} + \frac{(\log(2n))^3}{\bar{\epsilon}}\right) = \tilde{O}\left(\sqrt{\frac{n}{\epsilon}} + \frac{1}{\epsilon}\right).$$

6 The adaptive two-sided lower bound

We now prove that the polynomial dependence on n cannot disappear entirely. The construction uses two abelian groups with almost equal orders and the same exponent¹⁴. A short adaptive transcript sees only a small system of formal linear equations, and with high probability the same equality pattern occurs in both groups.

Theorem 6.1 (Adaptive two-sided lower bound). *There are absolute constants $c, \epsilon_0 > 0$ and infinitely many integers n such that, for every $1/n^2 \leq \epsilon \leq \epsilon_0$, every adaptive randomized two-sided tester for the group property requires*

$$\Omega\left(n^{1/5} + \frac{1}{\epsilon}\right)$$

queries.

6.1 The two almost-equal groups

By Maynard's theorem Theorem 2.7, infinitely many consecutive prime pairs $p < q$ satisfy $q - p \leq 600$. Fix one such pair and consider the following groups additively:

$$H := \mathbb{F}_p^3 \times \mathbb{F}_q^2, \quad K := \mathbb{F}_p^2 \times \mathbb{F}_q^3. \quad (9.1)$$

Set $m := |H| = p^3 q^2$, and $n := |K| = p^2 q^3$. Then

$$\frac{m}{n} = \frac{p}{q}, \quad \rho := \frac{n - m}{n} = \frac{q - p}{q} = O(1/p). \quad (9.2)$$

Both groups have exponent pq , but $m \nmid n$.

The YES distribution chooses a uniform bijection $\pi_K : K \rightarrow [n]$ and transports the addition of K to $[n]$. The NO distribution chooses a uniform m -subset $S \subseteq [n]$, a uniform bijection $\pi_H : H \rightarrow S$, and uses the addition of H on S^2 . Outside S^2 define, say, $x * y := 1$. The outside rule will play no role once we condition on a transcript remaining inside S .

Lemma 6.2 (Every NO instance is constantly far). *For all sufficiently large bounded-gap prime pairs $p < q$, every operation in the support of the NO distribution is $1/6$ -far from every group operation on $[n]$.*

Proof. Fix a group operation $\diamond$ on $[n]$. If H were isomorphic to a subgroup of $([n], \diamond)$, Lagrange's theorem would imply $m \mid n$, contrary to our choice of m, n . Therefore the overlap theorem Theorem 2.6, applied to the injection $\pi_H : H \rightarrow ([n], \diamond)$, says that at most $7/9$ of the ordered pairs $(x, y) \in H^2$ satisfy $\pi_H(x + y) = \pi_H(x) \diamond \pi_H(y)$. Thus at least $2m^2/9$ entries inside S^2 disagree with $\diamond$, and

$$\text{dist}(*, \diamond) \geq \frac{2m^2}{9n^2} = \frac{2}{9} \left(\frac{p}{q}\right)^2.$$

Since $q - p \leq 600$, we have $p/q \rightarrow 1$ along the infinite sequence. The last quantity is therefore larger than $1/6$ for all sufficiently large pairs. $\square$

¹⁴Exponent is the minimum non-zero integer r such that $rg = 0$ for every element g in the group.

6.2 A coupling lemma for adaptive transcripts

The next lemma is more general than the particular groups above. Its proof is the heart of the lower bound.

Lemma 6.3 (Common-exponent transcript coupling). *Let*

$$H = \prod_{i=1}^r \mathbb{F}_{p_i}^{a_i}, \quad K = \prod_{i=1}^r \mathbb{F}_{p_i}^{b_i}$$

have the same exponent $P = p_1 \cdots p_r$, and orders $m < n$. Put

$$\rho := \frac{n - m}{n},$$

and

$$\kappa := \min_{V \in \{H, K\}, 1 \leq i \leq r} p_i^{d_i(V)},$$

where $d_i(H) = a_i$ and $d_i(K) = b_i$. Compare a random encoding of K on $[n]$ with a random encoding of H on a random m -subset of $[n]$. Every deterministic adaptive algorithm using at most $t \leq m/10$ queries has transcript distributions satisfying

$$d_{\text{TV}}(\text{Tr}_Y, \text{Tr}_N) \leq C \left(t\rho + \frac{t^2}{\kappa} \right) \quad (9.3)$$

for an absolute constant C .

Proof. Instead of sampling the YES and NO transcripts independently, we generate them together, using the same random choices whenever possible. This is called *coupling*. If the coupling is successful, then $\text{Tr}_Y = \text{Tr}_N$. The two transcripts differ only if one of a small number of bad events occurs.

This is useful because, for any coupling of the two transcript distributions,

$$d_{\text{TV}}(\text{Tr}_Y, \text{Tr}_N) \leq \Pr[\text{Tr}_Y \neq \text{Tr}_N].$$

First we describe a symbolic version of the interaction that does not depend on whether the hidden group is H or K . Second, we show that the symbolic transcript can be coupled with an actual interaction with either group, and that the two transcripts differ with probability at most $O(t^2/\kappa)$. Finally, in the NO case, we show that the probability that, during the first t queries, the tester encounters a label corresponding to an element outside the hidden subset S is $O(t\rho)$.

1. The symbolic transcript.

Recall that $P = \prod_i p_i$ is the common exponent of H and K , and that each of the two groups has the form

$$V = \prod_i \mathbb{F}_{p_i}^{d_i(V)}, \quad V \in \{H, K\}.$$

We use additive notation for these groups.

Let $R := \mathbb{Z}/P\mathbb{Z}$. If $a \in R$ and $v = (v_i)_i \in V = \prod_i \mathbb{F}_{p_i}^{d_i(V)}$, define $av := ((a \bmod p_i)v_i)_i$. Thus coefficients in R can be used to form linear combinations of elements of V . That is, V is an R -module. In particular, if $a_1, \dots, a_s \in R$ and $\xi_1, \dots, \xi_s \in V$, then $a_1\xi_1 + \dots + a_s\xi_s$ is a well-defined element of V .

We now describe a symbolic version of the tester's interaction with the multiplication table. The symbolic process simulates the interaction of the original deterministic tester. The tester itself continues to see labels in $[n]$, while the symbolic process associates with every label u that has appeared a *symbolic value* $\sigma(u) \in R^s$ where $s = 2t$. The symbolic values keep track of the

algebraic relations forced by the queries. In particular, if the symbolic value of a new answer is equal to the symbolic value already assigned to a previously seen label w , then the simulation uses the same label w as the answer.

A *visible label* is an element of $[n]$ that appears in the transcript, either as an operand of a query or as an answer returned by the oracle. Other elements of $[n]$ that do not appear in the transcript are called *unused labels*. As the interaction proceeds, we assign to every visible label $u \in [n]$ that has appeared a *symbolic value* $\sigma(u) \in R^s$. Some symbolic values are introduced as new formally *independent values*, while the remaining symbolic values are determined from previously introduced ones and are therefore R -linear combinations of them. Suppose that so far t formally independent symbolic values have been introduced. We represent them by the standard basis vectors $e_1, \dots, e_r \in R^s$.

Whenever the deterministic tester uses, as an operand of a query, a visible label u that has not appeared earlier in the transcript, we assign to u a new standard basis vector. If r such labels have already been introduced, we set $\sigma(u) := e_{r+1} \in R^s$. Thus the new vector means: “the hidden group element corresponding to this visible label is a fresh unknown element, independent of the algebraic relations already forced by the transcript.”

Now suppose the tester makes a query (u, v) , and suppose the symbolic values assigned to the two operands are $\sigma(u) = \alpha \in R^s$ and $\sigma(v) = \beta \in R^s$. Since the hidden groups are written additively, the symbolic value of the answer is $\gamma := \alpha + \beta \in R^s$.

If some previously seen label $w \in [n]$ satisfies $\sigma(w) = \gamma$, then the symbolic simulation uses w as the answer to the query. Otherwise, the symbolic simulation chooses uniformly an unused label $w \in [n]$, uses w as the answer, and sets $\sigma(w) := \gamma$.

In this way, at every stage there is a one-to-one correspondence between the symbolic vectors that have appeared and the visible labels used to represent them. Two symbolic values are declared equal exactly when their vectors are equal in R^s . For example, if two fresh operand labels have symbolic values $(1, 0)$ and $(0, 1)$, then the answer to their query has symbolic value $(1, 1)$. If the first label is then added to this answer, the resulting symbolic value is $(2, 1)$.

Each query has at most two operands, so after t queries at most $s = 2t$ new independent symbolic values have been introduced. Each query can introduce at most two new operand labels and one new answer label. Hence at most $3t$ distinct symbolic vectors have appeared.

Fix from now on one complete realization of this symbolic transcript, including all the fresh visible labels chosen by the symbolic process.

2. Realizing the symbolic transcript in a hidden group.

Fix one of the two groups $V \in \{H, K\}$. Suppose, without loss of generality, that the symbolic process introduced $s = 2t$ independent symbolic values. We now assign actual elements $\xi_1, \dots, \xi_s \in V$ to them.

Given a symbolic vector $\alpha = (a_1, \dots, a_s) \in R^s$, define its *evaluation in V* by $\alpha(\xi) := a_1\xi_1 + \dots + a_s\xi_s \in V$. Thus, for example, $(1, 0)(\xi) = \xi_1$, $(0, 1)(\xi) = \xi_2$, and $(1, 1)(\xi) = \xi_1 + \xi_2$. This definition has the important property $(\alpha + \beta)(\xi) = \alpha(\xi) + \beta(\xi)$. Therefore, the symbolic rule “the answer vector is $\alpha + \beta$ ” agrees exactly with the group operation in V .

We first consider an auxiliary way to choose $\xi_1, \dots, \xi_s$. Choose them sequentially without replacement from V . Thus, when ξ_j is chosen, it is uniform on $A_j := V \setminus \{\xi_1, \dots, \xi_{j-1}\}$. Equivalently, $(\xi_1, \dots, \xi_s)$ is a uniformly random ordered tuple of distinct elements of V . We compare this auxiliary experiment with the true lazy exposure of a uniformly random labeled copy of V .

By *lazy exposure* we mean that we do not choose the whole random labeling in advance. Instead, whenever a previously unseen visible label first occurs, we reveal the hidden group element assigned to it. Since the labeling is a bijection, a new visible label must be assigned uniformly among the group elements that have not already been assigned to previously seen visible labels.

Consider the moment when the j -th new independent symbolic value is introduced. In the

auxiliary experiment, its actual value ξ_j is chosen uniformly from $A_j = V \setminus \{\xi_1, \dots, \xi_{j-1}\}$. In the true lazy exposure, however, the corresponding visible label must avoid not only the preceding independent values $\xi_1, \dots, \xi_{j-1}$, but every group element already assigned to a visible label.

Let $E_{j-1} \subseteq V$ denote the set of all hidden group elements already exposed immediately before the j -th independent value is introduced. The true lazy exposure therefore chooses the new hidden value uniformly from $B_j := V \setminus E_{j-1}$.

Condition on the event that the auxiliary experiment and the true lazy exposure have agreed up to the moment when the j -th independent label is introduced. The auxiliary experiment chooses its next value uniformly from $A_j := V \setminus \{\xi_1, \dots, \xi_{j-1}\}$, whereas the true lazy exposure chooses uniformly from $B_j := V \setminus E_{j-1}$. Since the previously chosen values $\xi_1, \dots, \xi_{j-1}$ have already been exposed, $B_j \subseteq A_j$.

We couple the two choices as follows. Choose the auxiliary value uniformly from A_j . If it belongs to B_j , use the same value in the true experiment. Thus the two choices can disagree only when the auxiliary value lies in $A_j \setminus B_j$. Consequently,

$$\begin{aligned} \Pr[\text{disagreement at step } j \mid \text{agreement at all previous steps}] &\leq \frac{|A_j \setminus B_j|}{|A_j|} \leq \frac{|E_{j-1}|}{|V| - 2t} \\ &\leq \frac{3t}{|V| - 2t} = O\left(\frac{t}{|V|}\right). \end{aligned}$$

There are at most $2t$ independent values to expose. Hence, by a union bound over the exposure steps, the probability that the two experiments ever disagree during this part of the coupling is

$$O\left(\frac{t^2}{|V|}\right) \leq O\left(\frac{t^2}{\kappa}\right). \quad (6.2.1)$$

A second possible source of disagreement remains. Two different symbolic vectors may evaluate to the same actual element of V . We now bound the probability of such an *accidental equality*.

Let $\alpha, \beta \in R^s$, $\alpha \neq \beta$, be two different symbolic vectors that appear in the transcript. Let $\delta := \alpha - \beta$. Since $\delta \neq 0$ in R^s , there is some coordinate j for which the coefficient $\delta_j \in R$ is nonzero. Because $R = \mathbb{Z}/P\mathbb{Z}$, $P = \prod_i p_i$, the condition $\delta_j \neq 0$ in R implies that for at least one prime $p_i \mid P$, $\delta_j \not\equiv 0 \pmod{p_i}$.

Condition on all assigned values ξ_r , $r \neq j$. The accidental equality $\alpha(\xi) = \beta(\xi)$ is equivalent to $\delta(\xi) = 0$. Looking only at the $\mathbb{F}_{p_i}^{d_i(V)}$ -component, this equation has the form $\delta_j(\xi_j)_i = w$, where $w \in \mathbb{F}_{p_i}^{d_i(V)}$ is completely determined by the already conditioned variables. Since $\delta_j \not\equiv 0 \pmod{p_i}$, multiplication by δ_j is invertible in $\mathbb{F}_{p_i}$. Hence the equation determines the entire $\mathbb{F}_{p_i}^{d_i(V)}$ -component of ξ_j uniquely.

There are $p_i^{d_i(V)}$ possible values for this component. Thus, among all elements of V , at most $|V|/p_i^{d_i(V)}$ can satisfy the equality. Under the auxiliary experiment, after conditioning on all other ξ_r 's, the variable ξ_j is uniform on a set of at least $|V| - 2t$ available elements. Since $t \leq |V|/10$, we obtain

$$\Pr[\alpha(\xi) = \beta(\xi)] \leq \frac{|V|/p_i^{d_i(V)}}{|V| - 2t} \leq \frac{2}{p_i^{d_i(V)}} \leq \frac{2}{\kappa}.$$

At most $3t$ symbolic vectors appear, so the number of unordered pairs of distinct symbolic vectors is at most

$$\binom{3t}{2} = O(t^2).$$

A union bound therefore gives

$$\Pr[\text{some accidental equality}] = O\left(\frac{t^2}{\kappa}\right). \quad (6.2.2)$$

Suppose neither the lazy-exposure coupling in (6.2.1) nor an accidental equality in (6.2.2) occurs. Then two visible labels correspond to the same hidden group element if and only if their symbolic vectors are equal. Consequently every query has exactly the equality pattern prescribed by the symbolic transcript.

Moreover, whenever the answer to a query is a group element that has not appeared earlier, a uniformly random labeling assigns to it a uniformly random unused visible label. This is exactly how the symbolic oracle chose its fresh answer labels. Thus the visible transcript can be coupled with the symbolic transcript so that they are identical, except with probability $O(t^2/\kappa)$. This argument applies to both $V = H$ and $V = K$.

3. The hidden subset in the NO experiment.

The preceding argument treats H and K in the same way as abstract hidden groups. There is one additional issue in the NO distribution: the copy of H occupies only a uniformly random subset $S \subseteq [n]$, $|S| = m$. The multiplication table agrees with the hidden group law only on $S \times S$.

Call a visible label *externally introduced* if its first appearance in the transcript is as an operand chosen by the tester, rather than as an answer returned by the oracle. These are exactly the labels to which the symbolic construction assigns new independent vectors. Since each of the t queries has two operands, there are at most $2t$ externally introduced labels.

Assume that all externally introduced labels seen so far belong to S . Then every oracle-generated answer also belongs to S , because the product of two elements of the hidden copy of H is again an element of that copy.

Let $T \subseteq S$ be the set of visible labels in S that have already been exposed. Conditional on the information revealed so far, the hidden set S is uniform among all m -subsets of $[n]$ that contain T . Consider a new externally introduced visible label $u \in [n] \setminus T$. For a uniformly random such hidden set S ,

$$\Pr[u \notin S \mid T] = \frac{n - m}{n - |T|}.$$

Since at most $3t$ visible labels have appeared, $|T| \leq 3t$. Under our assumption $t \leq n/10$, $n - |T| \geq n - 3t \geq 7n/10$. Therefore

$$\Pr[u \notin S \mid T] \leq 2 \frac{n - m}{n} = 2\rho.$$

There are at most $2t$ externally introduced labels, so a union bound gives

$$\Pr[\text{some externally introduced label lies outside } S] = O(t\rho). \quad (6.2.3)$$

It remains to check that oracle-generated fresh labels have the same distribution in the NO and YES experiments. Suppose the hidden group operation in H produces a group element that has not yet been exposed. Conditional on the current exposed set $T \subseteq S$, its visible label is uniform on $S \setminus T$. Now average also over the posterior distribution of the random set S .

Fix an unused visible label $u \in [n] \setminus T$. The probability that u belongs to S is $(m - |T|)/(n - |T|)$. Conditional on $u \in S$, the probability that u is the visible label assigned to the new hidden group element is $1/(m - |T|)$. Hence

$$\Pr[u \text{ is the next oracle-generated fresh label} \mid T] = \frac{m - |T|}{n - |T|} \cdot \frac{1}{m - |T|} = \frac{1}{n - |T|}.$$

Thus the next fresh oracle-generated label is uniform on all unused visible labels $[n] \setminus T$. This is exactly the same distribution as in the YES experiment and in the symbolic transcript.

After such a label is exposed, the conditional distribution of S is again uniform among all m -subsets containing the enlarged exposed set. Hence the same argument may be repeated throughout the interaction.

4. Conclusion.

We now couple the YES and NO experiments through the same symbolic transcript. The two visible transcripts are identical unless at least one of the following events occurs:

1. the auxiliary realization and the true lazy exposure fail to couple;
2. two distinct symbolic vectors accidentally evaluate to the same hidden group element;
3. in the NO experiment, an externally introduced visible label falls outside the hidden subset S .

For each of H and K , the first two events have total probability $O(t^2/\kappa)$ by (6.2.1) and (6.2.2). The third event has probability $O(t\rho)$ by (6.2.3). Absorbing the constant number of such terms into one absolute constant C , we obtain a coupling for which

$$\Pr[\text{Tr}_Y \neq \text{Tr}_N] \leq C \left(t\rho + \frac{t^2}{\kappa} \right).$$

By the coupling characterization of total variation distance,

$$d_{\text{TV}}(\text{Tr}_Y, \text{Tr}_N) \leq \Pr[\text{Tr}_Y \neq \text{Tr}_N] \leq C \left(t\rho + \frac{t^2}{\kappa} \right),$$

as claimed. □

6.3 Finishing the $n^{1/5}$ lower bound

The $\Omega(n^{1/5})$ lower bound. For the groups in (9.1), the smallest primary component is $\kappa = \min\{p^3, q^2, p^2, q^3\} = p^2$. Together with (9.2), the coupling lemma gives

$$d_{\text{TV}}(\text{Tr}_Y, \text{Tr}_N) \leq C' \left(\frac{t}{p} + \frac{t^2}{p^2} \right). \tag{9.6}$$

Choose $t = c_0 p$ for a sufficiently small absolute constant $c_0 > 0$. Then the right side is below $1/3$.

To translate this indistinguishability statement into a testing lower bound, use Yao's minimax principle: Fix equal prior probabilities on the YES and NO distributions. For any deterministic t -query algorithm, let Tr_Y and Tr_N denote its transcript distributions in the two cases. The maximum possible probability of distinguishing these two transcript distributions is $(1 + d_{\text{TV}}(\text{Tr}_Y, \text{Tr}_N))/2$. Consequently, an algorithm that succeeds with probability at least $2/3$ must satisfy $d_{\text{TV}}(\text{Tr}_Y, \text{Tr}_N) \geq 1/3$. Therefore, if every deterministic t -query algorithm satisfies $d_{\text{TV}}(\text{Tr}_Y, \text{Tr}_N) < \frac{1}{3}$, then, by Yao's minimax principle, no randomized t -query tester can distinguish the YES and NO cases with success probability at least $2/3$. Here every YES input is a group, and by Lemma 6.2 every NO input is $1/6$ -far from every group. Hence $\Omega(p)$ adaptive queries are necessary at constant distance.

Finally, $q - p \leq 600$ implies $n = p^2 q^3 = \Theta(p^5)$, so $p = \Theta(n^{1/5})$. Thus the constant-distance lower bound is $\Omega(n^{1/5})$.

The $\Omega(1/\epsilon)$ lower bound. We also need the standard $\Omega(1/\epsilon)$ term. In our setting this follows from a simple direct argument.

Fix any group table M on $[n]$, and fix a label, say $1 \in [n]$. Recall that every row of every group table is a permutation of $[n]$. In particular, every row of a group table contains the value 1 exactly once. Let $k := \lceil \epsilon n^2 \rceil$. We construct a random table M' by changing randomly chosen entries of M to the value 1. We choose only entries whose value in M is different from 1, so every chosen entry is actually changed. If $k \leq n$, choose k rows uniformly without replacement. In each chosen row, choose two positions uniformly among the $n - 1$ positions whose value in M is not 1, and change both entries to 1. In each chosen row, M' therefore contains the value 1 in at least two of the chosen positions. Since every row of every group table contains 1 only once, any group table must disagree with M' in at least one of these two positions. Hence every group table disagrees with M' in at least $k \geq \epsilon n^2$ entries. If $k > n$, let

$$L := \left\lceil \frac{k}{n} \right\rceil + 1.$$

In every row, choose L positions uniformly among those whose value in M is not 1, and change all of them to 1. Any group table can contain 1 in only one position in that row, and hence must disagree with M' in at least $L - 1$ of the chosen positions. Therefore the total number of disagreements is at least

$$n(L - 1) = n \left\lceil \frac{k}{n} \right\rceil \geq k \geq \epsilon n^2.$$

Thus in both cases every table M' in the support of the NO distribution is ϵ -far from every group table. It remains to show that few queries are unlikely to find one of the changed entries. For $1/n^2 \leq \epsilon \leq \epsilon_0$, where $\epsilon_0 > 0$ is a sufficiently small absolute constant, every fixed entry of the table is changed with probability $O(\epsilon)$. Indeed, in the first case this probability is at most

$$\frac{k}{n} \cdot \frac{2}{n - 1} = O(\epsilon),$$

and in the second case it is at most $L/(n - 1) = O(\epsilon)$.

Now fix a deterministic adaptive tester making t queries. Until the tester queries a changed entry, all answers it receives from M' are exactly the same as the answers it receives from M . Consequently, before the first changed entry is found, the tester follows exactly the same sequence of queries that it would follow on the fixed group table M . In particular, this sequence is independent of the random locations of the changed entries. By a union bound,

$$\Pr[\text{the tester queries a changed entry}] = O(t\epsilon).$$

Hence we may couple the YES input M and the random NO input M' so that their transcripts are identical except with probability $O(t\epsilon)$. Therefore $d_{\text{TV}}(\text{Tr}_Y, \text{Tr}_N) \leq O(t\epsilon)$.

A tester that distinguishes the YES and NO cases with probability at least $2/3$ requires total variation distance at least $1/3$. Thus $t = \Omega(1/\epsilon)$.

Combining the two bounds gives

$$\Omega\left(n^{1/5} + \frac{1}{\epsilon}\right).$$

7 Conclusion

For adaptive two-sided testing, the paper gives two complementary upper bounds. The first is an efficient tester using

$$O\left(n + \frac{1}{\epsilon}\right)$$

queries and $\tilde{O}(n + 1/\epsilon)$ time. The second gives the stronger query bound

$$\tilde{O}\left(\sqrt{\frac{n}{\epsilon}} + \frac{1}{\epsilon}\right)$$

when arbitrary finite postprocessing is allowed. Its central idea is to use expansion twice: first to make a short locator hit a random dictionary, and then to show that local consistency with the generator action forces the locator map to be nearly bijective.

The lower bound

$$\Omega\left(n^{1/5} + \frac{1}{\epsilon}\right)$$

holds along an infinite sequence of input sizes and shows that a polynomial gap remains. At constant distance the current bounds are, up to logarithmic factors,

$$\Omega(n^{1/5}) \quad \text{and} \quad O(\sqrt{n}).$$

Closing this gap is the main open problem suggested by the results.

AI Use and Author Responsibility

I take full responsibility for everything in my paper. Where I used AI assistance, I carefully considered its suggestions and ensured that the results, proofs, references, and other claims were correct and accurate. My role went beyond checking its answers. In many cases, I rejected its suggestions and proposed other approaches to improve the results. Using a tool for parts of my paper does not reduce my responsibility for mistakes, missing details, or failures to give proper credit to previous work.

References

- [1] V. Acciario and M. D. Atkinson. A new algorithm for testing the regularity of a permutation group. *Congressus Numerantium*, 90:151–160, 1992.
- [2] N. Alon and Y. Roichman. Random Cayley graphs and expanders. *Random Structures & Algorithms*, 5(2):271–284, 1994.
- [3] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549–595, 1993.
- [4] N. H. Bshouty. Sublinear time algorithms for abelian group property testing. In *51st International Symposium on Mathematical Foundations of Computer Science (MFCS 2026)*, volume 386 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 89:1–89:14, 2026.
- [5] Nader H. Bshouty and Oded Goldreich. On properties that are non-trivial to test. In Oded Goldreich, editor, *Computational Complexity and Local Algorithms - On the Interplay Between Randomness and Computation*, volume 15700 of *Lecture Notes in Computer Science*, pages 383–387. Springer, 2025.
- [6] F. Ergün, S. Kannan, S. R. Kumar, R. Rubinfeld, and M. Viswanathan. Spot-checkers. *Journal of Computer and System Sciences*, 60(3):717–751, 2000.
- [7] S. Evra, S. Gadot, O. Klein, and I. Komargodski. Verifying groups in linear time. In *Proceedings of the 65th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 2131–2147, 2024.

- [8] Eldar Fischer. A basic lower bound for property testing. *CoRR*, abs/2403.04999, 2024.
- [9] K. Friedl, G. Ivanyos, and M. Santha. Efficient testing of groups. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 157–166, 2005.
- [10] O. Goldreich and L. Tauber. On testing group properties. In *Computational Complexity and Local Algorithms*, volume 15700 of *Lecture Notes in Computer Science*, pages 431–450. Springer, 2025.
- [11] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.
- [12] G. Ivanyos, F. Le Gall, and Y. Yoshida. On the distance between non-isomorphic groups. *European Journal of Combinatorics*, 33(4):474–476, 2012.
- [13] Z. Landau and A. Russell. Random Cayley graphs are expanders: A simple proof of the Alon–Roichman theorem. *Electronic Journal of Combinatorics*, 11(1):R62, 2004.
- [14] J. Maynard. Small gaps between primes. *Annals of Mathematics*, 181(1):383–413, 2015.
- [15] L. Pyber. Enumerating finite groups of given order. *Annals of Mathematics*, 137(1):203–220, 1993.
- [16] S. Rajagopalan and L. J. Schulman. Verification of identities. *SIAM Journal on Computing*, 29(4):1155–1163, 2000.

A Background on Random Cayley Graphs

This appendix explains the spectral terminology used in Section 2.3 and derives Theorem 2.3 and Corollary 2.4 from the Alon–Roichman theorem [2] and the quantitative version of Landau and Russell [13].

A.1 Cayley multigraphs and their transition matrices

Let H be a finite group of order n , and let $a_1, \dots, a_k \in H$. We use the symmetric multiset

$$\Sigma := \{a_1, a_1^{-1}, \dots, a_k, a_k^{-1}\}.$$

The *right Cayley multigraph* has vertex set H . From every $x \in H$, and for every occurrence of $a \in \Sigma$, it has the edge $x \rightarrow xa$. Because Σ is symmetric, every edge occurs together with its reverse.

The *normalized transition matrix* P is the $n \times n$ matrix whose entry $P(x, y)$ is the probability that one step of the random walk starting from x ends at y . Explicitly,

$$P(x, y) := \frac{|\{a \in \Sigma : xa = y\}|}{|\Sigma|},$$

where elements of the multiset are counted with multiplicity.

We regard a function $f : H \rightarrow \mathbb{R}$ as a vector indexed by the elements of H . Then $(Pf)(x)$ denotes the x -coordinate of the vector obtained by multiplying f by P . Since $|\Sigma| = 2k$, we may equivalently write

$$(Pf)(x) = \frac{1}{2k} \sum_{i=1}^k (f(xa_i) + f(xa_i^{-1}))$$

for every real-valued function $f : H \rightarrow \mathbb{R}$.

Because the multiset is symmetric, $P(x, y) = P(y, x)$. Thus P is a real symmetric matrix.

Moreover, every row of P sums to 1, so the constant vector $\mathbf{1} = (1, \dots, 1)^T$ satisfies $P\mathbf{1} = \mathbf{1}$. Hence 1 is an eigenvalue of P .

A.2 The 2-norm and the 2-to-2 operator norm

For a vector $v = (v_1, \dots, v_n) \in \mathbb{R}^n$, its Euclidean norm is

$$\|v\|_2 := \left(\sum_{i=1}^n v_i^2 \right)^{1/2}.$$

For an $n \times n$ real matrix A , the *2-to-2 operator norm* is

$$\|A\|_{2 \rightarrow 2} := \max_{v \neq 0} \frac{\|Av\|_2}{\|v\|_2}.$$

If A is real and symmetric, then all its eigenvalues are real and $\|A\|_{2 \rightarrow 2} = \max_i |\lambda_i(A)|$. Thus for symmetric matrices the operator norm is simply the largest absolute value of an eigenvalue.

A.3 Why $P - J/n$ removes the trivial eigenvalue

Let J denote the $n \times n$ all-ones matrix. For every vector $v \in \mathbb{R}^n$,

$$\frac{J}{n}v = \left(\frac{1}{n} \sum_{x \in H} v_x \right) \mathbf{1}.$$

Thus J/n is the orthogonal projection onto the one-dimensional space of constant vectors. Every vector $v \in \mathbb{R}^n$ can be written uniquely as $v = c\mathbf{1} + v_0$, where $v_0 \perp \mathbf{1}$. Since $P\mathbf{1} = \mathbf{1}$, and since P is symmetric, the subspace $\mathbf{1}^\perp$ is invariant under P . On the constant vectors, $(P - J/n)\mathbf{1} = 0$. On $\mathbf{1}^\perp$, $(J/n)v_0 = 0$, and therefore $(P - J/n)v_0 = Pv_0$. Consequently,

$$\left\| P - \frac{J}{n} \right\|_{2 \rightarrow 2}$$

is exactly the largest absolute value of an eigenvalue of P different from the trivial eigenvalue 1.

If the eigenvalues of P are $1 = \lambda_1, \lambda_2, \dots, \lambda_n$, we therefore have

$$\left\| P - \frac{J}{n} \right\|_{2 \rightarrow 2} = \max_{2 \leq i \leq n} |\lambda_i|. \quad (\text{A.3.1})$$

This is the quantity that Landau and Russell denote by the second eigenvalue of the Cayley graph.

The absolute values are important here. We need to control large negative eigenvalues as well as large positive ones.