Parallel Algorithms for Group Isomorphism via Code Equivalence

Michael Levet
Department of Computer Science, College of Charleston, SC, USA

Abstract

In this paper, we exhibit AC^3 isomorphism tests for coprime extensions $H \ltimes N$ where H is elementary Abelian and N is Abelian; and groups where $\mathrm{Rad}(G) = Z(G)$ is elementary Abelian and $G = \mathrm{Soc}^*(G)$. The fact that isomorphism testing for these families is in P was established respectively by Qiao, Sarma, and Tang (STACS 2011), and Grochow and Qiao (CCC 2014, *SIAM J. Comput.* 2017).

The polynomial-time isomorphism tests for both of these families crucially leveraged *small* (size $O(\log |G|)$) instances of Linear Code Equivalence (Babai, SODA 2011). Here, we combine Luks' group-theoretic method for Graph Isomorphism (FOCS 1980, *J. Comput. Syst. Sci.* 1982) with the fact that G is given by its multiplication table, to implement the corresponding instances of Linear Code Equivalence in AC^3.

As a byproduct of our work, we show that isomorphism testing of arbitrary central-radical groups is decidable using AC circuits of depth $O(\log^3 n)$ and size $n^{O(\log \log n)}$. This improves upon the previous bound of $n^{O(\log \log n)}$-time due to Grochow and Qiao (*ibid.*).

2012 ACM Subject Classification Theory of computation → Circuit complexity; Theory of computation → Parallel algorithms; Mathematics of computing → Combinatorial algorithms

Keywords and phrases Group Isomorphism, Circuit Complexity, Code Equivalence

Digital Object Identifier 10.4230/LIPIcs.SWAT.2026.30

Related Version *Full Version*: `https://arxiv.org/abs/2604.13953`

Funding This work was partially supported by CRC 358 Integral Structures in Geometry and Number Theory at Bielefeld and Paderborn, Germany; the Department of Mathematics at Colorado State University; James B. Wilson's NSF grant DMS-2319370; and travel support from the Department of Computer Science at the College of Charleston.

Acknowledgements I wish to thank Peter Brooksbank, Edinah Gnang, Joshua A. Grochow, Takunari Miyazaki, and James B. Wilson for helpful discussions, as well as the anonymous referees for their helpful feedback. Parts of this work began at Tensors Algebra-Geometry-Applications (TAGA) 2024. I wish to thank Elina Robeva, Christopher Voll, and James B. Wilson for organizing this conference.

1 Introduction

The Group Isomorphism problem (GpI) takes as input two finite groups G and H, and asks if there exists an isomorphism $\varphi : G \to H$. When the groups are given by their multiplication (Cayley) tables, it is known that GpI belongs to $\mathsf{NP} \cap \mathsf{coAM}$. The generator-enumeration strategy has time complexity $n^{\log(n)+O(1)}$, where n is the order of the group [24, 44]. The parallel complexity of generator-enumeration has been gradually improved – see [20]. Algorithmically, the best known bound for GpI is $n^{(1/4)\log(n)+O(1)}$ [47, 41] (see [37, Sec. 2.2]). Even the substantial work on practical algorithms in more succinct input models (e.g., [13, 23, 12, 18]) still results in an $n^{\Theta(\log n)}$-time algorithm in the general case [55].

The Cayley model is impractical when working with computer algebra software. Instead, the groups are given by generators as permutations or matrices, or as black-boxes. For permutation groups, GpI belongs to NP [39]. For matrix groups or black-box groups, GpI

belongs to $\mathsf{Promise}\Sigma_2^p$ [9]; it is open whether GpI belongs to NP or coNP in such succinct models. In the past several years, there have been significant advances on algorithms with worst-case guarantees on the serial runtime for special cases of this problem – see [30, 21, 28] for a survey.

Key motivation for GpI comes from its relationship to the Graph Isomorphism problem (GI). When the groups are given verbosely by their Cayley tables, GpI $\leq_m^{\mathsf{AC}^0}$ GI [56]. There is no AC^0-Treduction from GI to GpI [19]. In light of Babai's breakthrough result that GI $\in$ QuasiP [4], GpI is a key barrier to improving the complexity of GI. There is considerable evidence suggesting that GI, and hence GpI, is not NP-complete [48, 17, 32, 4, 35, 1].

In this paper, we will investigate the parallel complexity of GpI. There are two key motivations for this. The first comes from GI, whose best known lower bound is DET [53], which is a subclass of TC^1. There is a large body of work spanning almost 40 years on NC algorithms for GI – see [38] for a survey. In contrast, the work on NC algorithms for GpI is very new (see [26] for a survey), and GpI is strictly easier than GI under AC^0-reductions [19]. So it is surprising that we know more about parallelizing subclasses of GI than subclasses of GpI. The second key motivation comes from the fact that complexity-theoretic lower bounds for GpI remain open, even against depth-2 AC circuits. In contrast, GpI is solvable using depth-4 $\mathsf{quasiAC}^0$ circuits [20]. Isomorphism testing is in P for a number of families of groups, and closing the gap between P and AC^0 for special cases is a key stepping stone in trying to characterize the complexity of the general GpI problem. We now turn to stating our main results.

Isomorphism Testing of Coprime Extensions. We will first recall the following notation [46]. Let $\mathcal{H}(\mathcal{A}, \mathcal{E})$ be the class of coprime extensions $H \ltimes N$, where N is Abelian and H is elementary Abelian.

▶ **Theorem 1** (cf. [46]). *There exists a uniform AC^3 algorithm that, given groups G_1, G_2 by their multiplication tables, decides if $G_1, G_2 \in \mathcal{H}(\mathcal{A}, \mathcal{E})$; and if so, decides if $G_1 \cong G_2$.*

There has been considerable work on polynomial-time isomorphism tests for coprime [36, 46, 8, 28, 14] and tame extensions [29]. In [46], polynomial-time isomorphism tests were given for coprime extensions $H \ltimes N$ where (i) H was $O(1)$-generated and N was Abelian, and (ii) $\mathcal{H}(\mathcal{A}, \mathcal{E})$. The parallel complexity for isomorphism testing of coprime extensions was first investigated by Grochow and Levet [28], who showed that some constant-dimensional Weisfeiler–Leman (WL) algorithm identifies, in $O(1)$-rounds, all coprime extensions $H \ltimes N$, where N is Abelian and H is $O(1)$-generated. Consequently, they obtained that the isomorphism problem for this family belongs to L. In his thesis, Brachter [14] showed that the more general class of groups with Abelian Sylow towers has WL-dimension $O(\log \log n)$. Consequently, Brachter's work yields $n^{O(\log \log n)}$-time bounds for isomorphism testing of this class. It remains open whether even the class $\mathcal{H}(\mathcal{A}, \mathcal{E})$ has bounded Weisfeiler–Leman dimension. Prior to this paper, it was open whether isomorphism testing of groups in $\mathcal{H}(\mathcal{A}, \mathcal{E})$ was in NC.

Methods. The polynomial-time isomorphism test for $\mathcal{H}(\mathcal{A}, \mathcal{E})$ [46] crucially leveraged *small* (size $O(\log |G|)$) instances of Linear Code Equivalence (CodeEq). Thus, in order to obtain our parallel bounds, we will establish the following.

▶ **Theorem 2.** *Let $m \in O(\log n)$, and let C_1, C_2 be linear codes of length m given by their generator matrices. We can decide equivalence between C_1 and C_2, as well as compute the coset of equivalences, using an AC circuit of depth $O((\log^2 n) \cdot \mathrm{poly}(\log \log n))$ and size $\mathrm{poly}(n)$.*

In establishing the bound of $(2+o(1))^m$-time for CodeEq, Babai reduces to isomorphism testing of graphs with $O(m)$ vertices [6]. When $m \in O(\log n)$, applying Babai's quasipolynomial GI procedure [4] would only yield bounds of $m^{O(\log^2 m)} = (\log n)^{O((\log\log n)^2)}$ runtime, which is not sufficient to obtain bounds of NC. Instead, we utilize Luks' group-theoretic method [40] to handle these small graphs. While Luks' claims of polynomial-time isomorphism testing are for graphs of bounded valence, the graphs we will consider need not satisfy this condition. However, our graphs are *small*, having only $O(\log n)$ many vertices. Thus, we bring to bear a standard suite of NC algorithms for permutation groups [3, 43]. As the group G is specified by its multiplication table and our graphs have $O(\log n) = O(\log |G|)$ many vertices, we can implement solutions for each subroutine of [3, 43] using an AC circuit of depth $\text{poly}(\log\log n)$ and size $\text{poly}(n)$. Grochow, Johnson, and Levet [26] introduced the notation $\mathsf{FOLL}^{O(1)}$ to refer to the class of languages decidable by uniform AC circuits of depth $\text{poly}(\log\log n)$ and size $\text{poly}(n)$. We combine this suite of $\mathsf{FOLL}^{O(1)}$permutation group algorithms, as well as the $\mathsf{FOLL}^{O(1)}$procedure for *small* instances of Coset Intersection (CosetInt) from [26].

Isomorphism Testing of Central-Radical Groups. We now turn to our second main result. Recall that the *solvable radical* is the unique maximal solvable normal subgroup of G. We will denote the solvable radical of G as $\text{Rad}(G)$. Here, we will consider *central-radical groups*, which are precisely those groups where the solvable radical $\text{Rad}(G) = Z(G)$.

▶ **Theorem 3** (cf. [30, Thm. C]). *Let G_1, G_2 be groups given by their multiplication tables. Suppose that G_1, G_2 have central, elementary Abelian radicals. Then isomorphism can be decided in AC^3, if either:*

(a) *$G_1/\text{Rad}(G_1)$ is a direct product of non-Abelian simple groups; or*

(b) *$G_1/\text{Rad}(G_1)$ is a direct product of perfect groups, each of order $O(1)$.*

This improves upon the previous polynomial-time bounds due to Grochow and Qiao [30].

Key motivation for isomorphism testing of central-radical groups comes from the following series of characteristic subgroups, known as the *Babai–Beals filtration* [5]: $1 \leq \text{Rad}(G) \leq \text{Soc}^*(G) \leq \text{PKer}(G) \leq G$. We will now explain the terms on this chain. Let $\pi : G \to G/\text{Rad}(G)$ be the natural projection map. $\text{Soc}^*(G)$ is the preimage of the socle[1] $\text{Soc}(G/\text{Rad}(G))$, under π. The group $\text{Soc}^*(G)/\text{Rad}(G) = \text{Soc}(G/\text{Rad}(G))$ is the direct product of non-Abelian simple groups $T_1, \ldots, T_k$. The conjugation action of G on $\text{Soc}(G/\text{Rad}(G))$ induces a permutation action φ on $T_1, \ldots, T_k$. Now $\text{PKer}(G) := \text{Ker}(\varphi)$. Note that Thm. 3(a) considers precisely those groups where $\text{Rad}(G) = Z(G)$ is elementary Abelian, and $G = \text{Soc}^*(G)$. While these assumptions might feel restrictive, Grochow and Qiao essentially leveraged the full weight of [30] in order to obtain a polynomial-time isomorphism test, combining deep mathematical insights (group cohomology) with demanding algorithmic techniques (see under Methods, below).

The class of *Fitting-free* groups consists precisely of those groups G where $\text{Rad}(G)$ is trivial. There has been considerable work on isomorphism testing of Fitting-free groups. In the Cayley table model, it took a series of two papers [6, 7] to obtain a polynomial-time isomorphism test, and this bound was only recently improved this bound from P to AC^3 [26]. Fitting-free groups have also received significant attention, from the perspective of the Weisfeiler–Leman algorithm [15, 14, 28, 27, 26].

[1] The *socle* of a group H is the direct product of the minimal normal subgroups of H.

In light of the efficient isomorphism tests for Fitting-free groups [6, 7, 26], the next step is to consider groups where $\mathrm{Rad}(G)$ is non-trivial. In [6], the authors posed the problem of handling isomorphism for central-radical groups. Grochow and Qiao [30] exhibited an $n^{O(\log\log n)}$-time algorithm to decide isomorphism of arbitrary central-radical groups, as well as groups where $\mathrm{Rad}(G)$ was elementary Abelian (not necessarily central). For additional complexity-theoretic work on isomorphism testing for groups where $\mathrm{Rad}(G)$ is non-trivial, see [29, 16, 37, 14].

Methods. We will now outline the key ideas and techniques involved in proving Thm. 3. As a conceptual starting point, let us first consider the setting of direct products. The Remak–Krull–Schmidt theorem provides that two direct products $G = G_1 \times \cdots \times G_k$ and $H = H_1 \times \cdots \times H_\ell$ are isomorpic if and only if $k = \ell$, and for some permutation $\sigma \in \mathrm{Sym}(k)$, $G_i \cong H_{\sigma(i)}$ for all $i \in [k]$. Grochow and Qiao established that for the groups in Thm. 3, an analogous result holds [30, Lem. 7.4] (recalled as Lem. 27).

Let G_1, G_2 be two groups under consideration in Thm. 3. For $i = 1, 2$, write $G_i/\mathrm{Rad}(G_i) = \prod_{j=1}^{k} T_{i,j}$. Let $\pi_i : G_i \to G_i/\mathrm{Rad}(G_i)$ be the natural projection map. Rather than considering the isomorphism type of the $T_{i,j}$, we will need to consider the isomorphism type of each $\pi_i^{-1}(T_{i,j})$. It is then necessary to determine whether there exists a permutation $\sigma \in \mathrm{Sym}(k)$ such that $\pi_1^{-1}(T_{1,j})$ and $\pi_2^{-1}(T_{2,\sigma(j)})$ are isomorphic. However, there is an additional complication, in that we require a single isomorphism $\alpha : Z(G_1) \cong Z(G_2)$, such that for all $j \in [k]$, α can be extended to an isomorphism of $\pi_1^{-1}(T_{1,j})$ and $\pi_2^{-1}(T_{2,\sigma(j)})$. Grochow and Qiao handled this step using *small* (size $O(\log|G|)$) instances of CodeEq and CosetInt. In these places, we will use our parallel implementation for CodeEq (Thm. 2), as well as the $\mathsf{FOLL}^{O(1)}$procedure for *small* instances of CosetInt from [26].

Still, we will need an additional ingredient. In order to determine the isomorphism types of the $\pi_i^{-1}(T_{i,j})$ ($i = 1, 2$; $j \in [k]$), Grochow and Qiao utilized [30, Thm. 6.1]. Note that [30, Thm. 6.1] is quite powerful. Grochow and Qiao obtained one of their main results [30, Thm. A/Cor. 6.2] as a corollary: a bound of $n^{O(\log\log n)}$-time for isomorphism testing of central-radical groups, as well as the same bound for computing the coset of isomorphisms when $\mathrm{Rad}(G_i) = Z(G_i)$ was elementary Abelian. We will parallelize [30, Thm. 6.1]; for readability, we refer to Thm. 22, rather than stating the exact theorem here. As a byproduct, we obtain the following complexity-theoretic improvement in isomorphism testing of arbitrary central-radical groups.

▶ **Theorem 4** (cf. [30, Thm. A]). *Isomorphism of central-radical groups of order n, given by their Cayley tables, can be decided by AC circuits of depth $O(\log^3 n)$ and size $n^{O(\log\log n)}$.*

2 Preliminaries

2.1 Groups and Codes

Groups. All groups will be assumed to be finite. A *Hall* subgroup of a group G is a subgroup N such that $|N|$ and $|G/N|$ are coprime. When a Hall subgroup is normal, we refer to G as a coprime extension. We will consider the following classes of finite groups, using the notation of Qiao, Sarma, and Tang [46]. Let $\mathcal{E}$ be the class of elementary Abelian groups, and $\mathcal{A}$ the class of Abelian groups. For classes of finite groups $\mathcal{X}, \mathcal{Y}$, let $\mathcal{H}(\mathcal{X}, \mathcal{Y})$ denote the class of coprime extensions $H \ltimes N$, where $N \in \mathcal{X}$ and $H \in \mathcal{Y}$.

Let G be a group. Given $g, h \in G$, the *commutator* $[g,h] := ghg^{-1}h^{-1}$. For sets $X, Y \subseteq G$, the *commutator subgroup* $[X,Y] := \langle\{[g,h] : g \in X, h \in Y\}\rangle$. We say that G is *perfect* if $G = [G,G]$, and that G is *centerless* if $Z(G) = 1$. Let $d(G)$ be the minimum size taken over all generating sets of G. A *basis* of an Abelian group A is a set of generators $\{a_1, \ldots, a_k\}$ such that $A = \langle a_1\rangle \times \cdots \times \langle a_k\rangle$.

Codes. Let $\mathbb{F}$ be a field. Let $\mathrm{Mat}_{n\times m}(\mathbb{F})$ denote the set of $n \times m$ matrices over the field $\mathbb{F}$. $\mathrm{GL}_m(\mathbb{F})$ denotes the set of $m \times m$ invertible matrices over $\mathbb{F}$. A *linear code* of length m is a subspace $U \leq \mathbb{F}^m$. A $d \times m$ matrix A over $\mathbb{F}$ *generates* the code U if the rows of A span U. Let U, W be d–dimensional codes of length m over $\mathbb{F}$, generated by $d \times m$ matrices A, B, respectively. Then U and V are *equivalent* if and only there exists a permutation matrix $P \in \mathrm{GL}_m(\mathbb{F})$ and a matrix $T \in \mathrm{GL}_d(\mathbb{F})$ such that $B = TAP$. If A_1, A_2 are generator matrices for two codes U_1, U_2 respectively, we write $\mathrm{CodeEq}(A_1, A_2)$ for the coset of code equivalences taking U_1 to U_2.

2.2 Group Extensions and Cohomology

We recall preliminaries concerning group extensions and Abelian cohomology from [30]. Given a finite group G, we will consider an Abelian normal subgroup $A \trianglelefteq G$ when considering G as an extension of A by $Q := G/A$. Here, we denote this as $A \overset{\iota}{\hookrightarrow} G \overset{\pi}{\twoheadrightarrow} Q$, where $\iota : A \to G$ is an injection and $\mathrm{Im}(\iota) = \ker(\pi)$. As A is Abelian, we write the group operation of A additively, even though we denote the group operation of G multiplicatively – this is standard in this area. Despite that $A \leq G$, we will use these notations in different contexts, and so it should not cause confusion. We refer to G as the *total group*.

Let $\pi : G \to G/A \cong Q$ be the natural projection map. Any function $s : Q \to G$ such that $\pi(s(q)) = q$ for all $q \in Q$ is called a *section* of π. Any such section gives rise to a function $f_s : Q \times Q \to A$ defined by $f_s(p,q) = s(p)s(q) \cdot s(pq)^{-1}$. We are free to choose $s(1) = \mathrm{id}_G$, and then $f(1,q) = f(q,1) = 0$ for all $q \in Q$. Such sections are called *normalized*. We will assume that all sections are normalized, unless otherwise stated.

For $g \in G$, the conjugation action $\theta_g : G \to G$ is defined by $\theta_g(x) = gxg^{-1}$. As the operation of G is associative, we have that for all $p, q, r \in Q$: $f_s(p,q) + f_s(pq,r) = \theta_p(f_s(q,r)) + f_s(p,qr)$. This is the 2-*cocycle identity*. Any function $f : Q \times Q \to A$ is called a 2-*cochain*. If f satisfies the 2-cocycle identity with respect to θ, then f is called a 2-*cocycle* with respect to θ. Given any homomorphism $\theta : Q \to \mathrm{Aut}(A)$, every 2-cocycle with respect to θ arises as f_s for some section s of some extension $A \hookrightarrow G \twoheadrightarrow Q$ with respect to θ. Given any function $u : Q \to A$, the 2-*coboundary* associated to u is the function $b_u : Q \times Q \to A$ defined by $b_u(p,q) = u(p) + \theta_p(u(q)) - u(pq)$. Any two 2-cocycles associated to the same extension differ by a coboundary. The 2-cochains form an Abelian group $C^2(Q,A)$ defined by pointwise addition: $(f+g)(p,q) = f(p,q) + g(p,q)$. Observe that the 2-cocycle identity is $\mathbb{Z}$-linear. Thus, the 2-cocycles form a subgroup of $C^2(Q,A)$, denoted by $Z^2(Q,A,\theta)$. Similarly, the 2-coboundaries form a subgroup of $Z^2(Q,A,\theta)$, denoted by $B^2(Q,A,\theta)$. A 2-*cohomology class* is a coset of $H^2(Q,A,\theta) := Z^2(Q,A,\theta)/B^2(Q,A,\theta)$. If $f \in Z^2(Q,A,\theta)$, we denote the corresponding cohomology class by $[f]$. It follows from the above discussion that each extension $A \hookrightarrow G \twoheadrightarrow Q$ determines a single cohomology class $[f] \in H^2(Q,A,\theta)$.

▶ **Definition 5** ([30, Def. 2.1]). *Let A be an Abelian group, and Q an arbitrary group. Let $\theta : Q \to Aut(A)$ be an action, and $f : Q \times Q \to A$ a 2-cocycle ($f \in Z^2(Q,A,\theta)$). The pair (θ, f) is called* extension data. *Two extension data for the pair (Q,A) are equivalent if they have the exact same action and if the two 2-cocycles are cohomologous.*

Given an extension $A \hookrightarrow G \twoheadrightarrow Q$, the associated extension data are the action θ as defined above, and any 2-cocycle f_s for any section $s : Q \to G$. The extension data are in general not unique – we may choose any representative of the corresponding 2-cohomology class. Furthermore, if the action is trivial, then this extension is called *central*. If the 2-cohomology class is trivial, then this extension is called *split*, and $G = P \ltimes A$ for some subgroup $P \leq G$ isomorphic to Q.

We now turn to discussing when two different extension data $(\theta_1, f_1), (\theta_2, f_2)$ yield isomorphic total groups. We first recall additional notation and terminology. Recall that a subgroup $N \leq G$ is *characteristic* if for all $\varphi \in \mathrm{Aut}(G)$, $\varphi(N) = N$. The analogous notion for isomorphisms (rather than automorphisms) is a function $\mathcal{S}$ that assigns to each group G a subgroup $\mathcal{S}(G)$ such that any isomorphism $\varphi : G_1 \to G_2$ restricts to an isomorphism $\varphi|_{\mathcal{S}(G_1)} : \mathcal{S}(G_1) \to \mathcal{S}(G_2)$. We call such a function $\mathcal{S}$ a *characteristic subgroup function*. Most natural characteristic subgroups encountered are characteristic subgroup functions – for instance, the center $Z(G)$, the commutator $[G, G]$, and the solvable radical $\mathrm{Rad}(G)$.

▶ **Definition 6** ([30, Definition 2.2]). *Let A be an Abelian group, and let Q be an arbitrary group. Let $(\theta_1, f_1), (\theta_2, f_2)$ be extension data for A-by-Q. Then the extension data are* pseudo-congruent *if there exists $(\alpha, \beta) \in Aut(A) \times Aut(Q)$ such that:*

$$\theta_1(q)(a) = \alpha^{-1}\Big(\theta_2(\beta(q))(\alpha(a))\Big) =: \theta_2^{\alpha,\beta}(q)(a). \tag{1}$$

for all $q \in Q, a \in A$; and $f_1(p, q) = \alpha^{-1}(f_2(\beta(p), \beta(q))) + b_u(p, q)$, for all $p, q \in Q$ and for some coboundary b_u. In this case, we write $(\theta_1, f_1) \cong (\theta_2, f_2)$.

▶ **Lemma 7** (see e.g., [30, Lem. 2.3] and [31, Sec. 2.7.4]). *Let $\mathcal{S}$ be a characteristic subgroup function. Let G_1, G_2 be groups, such that $\mathcal{S}(G_1)$ and $\mathcal{S}(G_2)$ are both Abelian. Then $G_1 \cong G_2$ if and only if both of the following conditions hold:*
1. *$\mathcal{S}(G_1) \cong \mathcal{S}(G_2)$ and $G_1/\mathcal{S}(G_1) \cong G_2/\mathcal{S}(G_2)$.*
2. *$(\theta_1, f_1) \cong (\theta_2, f_2)$, where (θ_i, f_i) is the extension data of $A \hookrightarrow G_i \twoheadrightarrow Q$ $(i = 1, 2)$.*

Note that if the 2-cohomolohy class is trivial (in which case, the extension splits), Lem. 7 yields Taunt's Lemma as a corollary:

▶ **Lemma 8** (Taunt [52]). *Let $G = H \ltimes_\theta N$ and $\widehat{G} = \hat{H} \ltimes_{\hat{\theta}} \widehat{N}$. If $\alpha : H \to \hat{H}$ and $\beta : N \to \widehat{N}$ are isomorphisms such that for all $h \in H$ and all $n \in N$, $\hat{\theta}_{\alpha(h)}(n) = (\beta \circ \theta_h \circ \beta^{-1})(n)$, then the map $(h, n) \mapsto (\alpha(h), \beta(n))$ is an isomorphism of $G \cong \widehat{G}$. Conversely, if G and $\widehat{G}$ are isomorphic and $|H|$ and $|N|$ are coprime, then there exists an isomorphism of this form.*

2.3 Computational Complexity

We assume that the reader is familiar with standard complexity classes such as $\mathsf{P}, \mathsf{NP}, \mathsf{L}$, and NL. Denote by FL the class of logspace computable functions. For a standard reference on circuit complexity, see [54]. We consider Boolean circuits using the gates AND, OR, NOT, and $\mathsf{Majority}$, where $\mathsf{Majority}(x_1, \ldots, x_n) = 1$ if and only if $\geq n/2$ of the inputs are 1. Otherwise, $\mathsf{Majority}(x_1, \ldots, x_n) = 0$. In this paper, we will consider $\mathsf{DTIME}(\log^c n)$-uniform circuit families $(C_n)_{n \in \mathbb{N}}$, for some fixed $c \geq 1$. For this, one encodes the gates of each circuit C_n by bit strings of length $O(\log^c n)$. Then the circuit family $(C_n)_{n \geq 0}$ is called $\mathsf{DTIME}(\log^c n)$-*uniform* if (i) there exists a deterministic Turing machine that computes for a given gate $u \in \{0, 1\}^*$ of C_n ($|u| \in O(\log^c n)$) in time $O(\log^c n)$ the type of gate u, where the types are $x_1, \ldots, x_n$, NOT, AND, OR, or $\mathsf{Majority}$ gates, and (ii) there exists a deterministic Turing

machine that decides for two given gates $u, v \in \{0,1\}^*$ of C_n ($|u|, |v| \in O(\log^c n)$) and a binary encoded integer i with $O(\log^c n)$ many bits in time $O(\log^c n)$ whether u is the i-th input gate for v. When $c = 1$, this notion of uniformity is referred to as DLOGTIME-uniformity. For circuit families of size $\text{poly}(n)$, we will use DLOGTIME-uniformity.

▶ **Definition 9.** *Fix $k \geq 0$. We say that a language L belongs to (uniform) NC^k if there exist a (uniform) family of circuits $(C_n)_{n\in\mathbb{N}}$ over the AND, OR, NOT gates such that the following hold: (i) The AND and OR gates take exactly 2 inputs. That is, they have fan-in 2; (ii) C_n has depth $O(\log^k n)$ and uses (has size) $n^{O(1)}$ gates. Here, the implicit constants in the circuit depth and size depend only on L; and (iii) $x \in L$ if and only if $C_{|x|}(x) = 1$.*

The complexity class AC^k is defined analogously as NC^k, except that the AND, OR gates are permitted to have unbounded fan-in. That is, a single AND gate can compute an arbitrary conjunction, and a single OR gate can compute an arbitrary disjunction. The complexity class TC^k is defined analogously as AC^k, except that our circuits are now permitted Majority gates of unbounded fan-in. We also allow circuits to compute functions by using multiple output gates. For every k, the following containments are well-known: $\mathsf{NC}^k \subseteq \mathsf{AC}^k \subseteq \mathsf{TC}^k \subseteq \mathsf{NC}^{k+1}$. We also have that $\mathsf{NC}^1 \subseteq \mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{AC}^1$.

The complexity class FOLL is the set of languages decidable by uniform AC circuits of depth $O(\log \log n)$ and polynomial-size [11]. We will use a slight generalization of this: we use $\mathsf{FOLL}^{O(1)}$ to denote the class of languages L decidable by uniform AC circuits of depth $O((\log \log n)^c)$ for some c that depends only on L [26]. It is known that $\mathsf{AC}^0 \subsetneq \mathsf{FOLL}^{O(1)} \subsetneq \mathsf{AC}^1$, the former by a simple diagonalization argument on top of Sipser's result [50], and the latter because the Parity function is in AC^1 but not $\mathsf{FOLL}^{O(1)}$ (nor any depth $o(\log n/\log \log n)$). $\mathsf{FOLL}^{O(1)}$ cannot contain any complexity class that can compute Parity, such as $\mathsf{TC}^0, \mathsf{NC}^1, \mathsf{L}$, or NL, and it remains open whether any of these classes contain $\mathsf{FOLL}^{O(1)}$.

We will also be interested in NC circuits of quasipolynomial size (i. e., $2^{O(\log^k n)}$ for some constant k). For a circuit class $\mathcal{C} \subseteq \mathsf{NC}$, the analogous class permitting a quasipolynomial number of gates is denoted quasi$\mathcal{C}$. Note that DLOGTIME uniformity does not make sense for quasiNC, as we cannot encode gate indices using $O(\log n)$ bits. Instead, we will use DPOLYLOGTIME-uniformity for quasiNC [10, 25].

2.4 Permutation Group Algorithms

We will consider the permutation group model, in which groups are specified succinctly by a sequence of permutations from $\text{Sym}(m)$. The computational complexity for this model will be measured in terms of m. We will first recall some key concepts concerning permutation groups – see [22] for a general reference. A permutation group $G \leq \text{Sym}(m)$ is *transitive* if the permutation domain $[m]$ is a single G-orbit. An equivalence relation $\sim$ on $[m]$ is G-invariant if $x \sim y \Leftrightarrow x^g \sim y^g$ for all $x, y \in [m], g \in G$. The discrete equivalence – in which all elements are pairwise inequivalent – and indiscrete equivalence – in which all elements are equivalent – are considered trivial. The equivalence classes of any G-invariant equivalence relation are called *blocks* of G. The singleton sets and the whole set $[m]$ are considered trivial blocks; any other blocks, if they exist, are non-trivial. G is *primitive* if it has no non-trivial blocks, or equivalently, has no non-trivial G-invariant equivalence relations.

Equivalently, a block for G is a subset $\Delta \subseteq [m]$ such that for every $g \in G$, Δ^g is either disjoint from Δ or equal to Δ. A *system of imprimitivity* for G is a collection of blocks that partition $[m]$ (equivalently, the equivalence classes of a G-invariant equivalence relation). If

G is transitive, every system of imprimitivity arises as $\{\Delta^g : g \in G\}$ for some block Δ. A block is *minimal* if it is non-trivial and contains no proper subset that is also a non-trivial block. Two distinct minimal blocks can intersect in at most one point.

We will now recall a standard suite of problems with known NC solutions in the setting of permutation groups. In our setting, m will be *small* ($m \in O(\text{polylog}(n))$) relative to the overall input size n, which will yield $\mathsf{FOLL}^{O(1)}$ bounds for these problems.

▶ **Lemma 10.** *Let $c \in \mathbb{Z}^+$, and let $m \in O(\log^c n)$. Let $G \leq Sym(m)$ be given by a sequence S of generators. The following problems are in $\mathsf{FOLL}^{O(1)}$ relative to n, that is, they have uniform AC circuits of depth $\text{poly}(\log\log n)$ and $\text{poly}(n)$ size:*

(a) *Compute the order of G.*
(b) *Decide whether a given permutation σ is in G; and if so, exhibit a word ω such that $\sigma = \omega(S)$.*
(c) *Find the kernel of any action of G.*
(d) *Find the pointwise stabilizer of $B \subseteq [m]$.*
(e) *Given a list of generators S for G, compute a minimal (non-redundant) of generators S' for G of size $\text{poly}(m)$.*
(f) *Computing a non-trivial, minimal block system for G, if one exists; otherwise, a report that G is primitive.*
(g) *Computing the orbits of G.*
(h) *CosetInt: Given $G, H \leq Sym(m)$ and $x, y \in Sym(m)$, compute $Gx \cap Hy$.*

Proof. Each of these problems in (a)–(g) is known to be solvable using a circuit of depth $O(\text{polylog}(m))$ and size $O(\text{poly}(m))$: see [3] for (a)–(e), [43] for (f)–(g). The $\mathsf{FOLL}^{O(1)}$ bound follows from the fact that $m \in O(\log^c n)$. For (h), the $\mathsf{FOLL}^{O(1)}$bound was established in [26]. ◀

Finally, we consider the Transversal problem, which takes as input $H < G \leq \text{Sym}(m)$ such that $[G : H] \in m^{O(1)}$ and asks for a left transversal for H in G. An NC algorithm was given when $|G| \in m^{O(\log m)}$ [34, Prop. 3.13]. We will instead consider Transversal when $m \in O(\log n)$ and $[G : H] \in (\log n)^{O(\log\log n)}$. A careful analysis of [34, Prop. 3.13] yields the following:

▶ **Lemma 11** (cf. [34, Prop. 3.13])**.** *Let $m \in O(\log n)$, and let $H < G \leq Sym(m)$. Suppose that $[G : H] \in (\log n)^{O(\log\log n)}$. We can solve Transversal using an AC circuit of depth $O((\log n) \cdot \text{poly}(\log\log n))$.*

2.5 Representation Theory of Finite Groups

We recall key preliminaries concerning the representation theory of finite groups. For a general reference, see [49]. Let G be a finite group, and let V be a vector space. A *representation* of G over V is a group homomorphism $\varphi : G \to \text{GL}(V)$. The *trivial representation* maps every group element to the identity matrix in $\text{GL}(V)$. If $V \cong \mathbb{F}^d$, for some field $\mathbb{F}$ and some $d \in \mathbb{N}$, a homomorphism $\varphi : G \to \text{GL}_d(\mathbb{F})$ is called a *representation* of G over $\mathbb{F}$ of dimension d. Let $\varphi : G \to \text{GL}(V)$ be a representation. A subspace $L \leq V$ is *invariant* with respect to φ if for all $g \in G$, $\varphi_g(L) = L$. Note that $0, V$ are the *trivial invariant subspaces*. A representation without any non-trivial invariant subspaces is called an *irreducible representation*.

Fix a field $\mathbb{F}$. For the rest of this section, we consider representations over $\mathbb{F}$. Let $\varphi : G \to \text{GL}(V)$ and $\rho : G \to \text{GL}(W)$ be representations. The *direct sum* $\varphi \oplus \rho$ is a representation of G over $V \oplus W$, defined as $(\varphi \oplus \rho)_g(u + v) = \varphi_g(u) + \rho_g(v)$, for all $g \in G$.

A representation is *completely reducible* if it is the direct sum of irreducible representations. Maschke's theorem states that if the characteristic of $\mathbb{F}$ is 0 or coprime to $|G|$, then any representation of G over $\mathbb{F}$ is completely reducible.

Two representations $\varphi, \psi : G \to \mathrm{GL}(V)$ are *equivalent* if there exists an invertible linear transformation $T : V \to V$ such that $\varphi(g) = T\psi(g)T^{-1}$ for all $g \in G$. Suppose that φ, ψ are completely reducible. Write $\varphi = \iota_1^{d_1} \oplus \cdots \oplus \iota_\ell^{d_\ell}$ and $\psi = \iota_1^{e_1} \oplus \cdots \oplus \iota_\ell^{e_\ell}$, where the ι_i's are irreducible and pairwise inequivalent, and each $d_i, e_i \geq 0$. We have that φ and ψ are equivalent if and only if $d_i = e_i$ for each $i \in [\ell]$.

▶ **Proposition 12.** *Let p, q be distinct primes. Let $G = \mathbb{Z}_q^\ell \ltimes_\varphi \mathbb{Z}_p^k$ be given by its multiplication table. In AC^1, we can list the irreducible components of φ and group them by equivalence type. In particular, each irreducible component is returned as a function of the form $\tau : \mathbb{Z}_q^\ell \to GL_{k_\tau}(\mathbb{Z}_p)$, where k_τ is the dimension of τ.*

3 Isomorphism Testing of Small Graphs in Parallel

In this section, we will establish the following.

▶ **Theorem 13** (cf. [40]). *Fix $n \in \mathbb{N}$. Let X, Y be graphs with $m \in O(\log n)$ vertices. We can decide whether X and Y are isomorphic, using an AC circuit of depth $O((\log n)^2 \cdot \mathrm{poly}(\log \log n))$ and size $\mathrm{poly}(n)$.*

In order to establish Thm. 13, we essentially parallelize the previous work of Luks for isomorphism testing for graphs of bounded valence [40]. While our graphs here need not have bounded valence, they are *small*. We will crucially take advantage of this, in tandem with a suite of efficient parallel algorithms for permutation groups (see Section 2.4).

Fix an edge e of X. We compute $\mathrm{Aut}_e(X)$, the automorphism group of X that fixes e. Following [40], define X_r to be the subgraph of X consisting of all vertices and all edges of X, which appear in paths of length $\leq r$ through e. So $X_1 = e$ and $X_{m-1} = X$. We will compute $\mathrm{Aut}_e(X_r)$ for each $r = 1, \ldots, m-1$. The groups are related via the homomorphisms: $\pi_r : \mathrm{Aut}_e(X_{r+1}) \to \mathrm{Aut}_e(X_r)$, where $\pi_r(\sigma)$ is the restriction of σ to X_r. Thus, given (generators for) $\mathrm{Aut}_e(X_r)$, determining generators for $\mathrm{Aut}_e(X_{r+1})$ reduces to two problems: find a set $\mathcal{R}$ of generators for $\mathrm{Ker}(\pi_r)$, and find a set $\mathcal{S}$ of generators for $\mathrm{Im}(\pi_r)$. A careful analysis of [40] shows that we can compute $\mathcal{R}$ in AC^0. It remains to compute $\mathcal{S}$.

▶ **Proposition 14.** *Take the same assumptions as Thm. 13 and fix $r \in [m-1]$. Suppose we are given generators for $Aut_e(X_r)$. We can compute $\mathcal{S}$ using an AC circuit of depth $O((\log n) \cdot \mathrm{poly}(\log \log n))$ and size $\mathrm{poly}(n)$.*

Prop. 14 immediately yields Thm. 13.

Proof of Thm. 13. We proceed for $m - 1 \in O(\log n)$ times iterations. At each iteration, we compute generators for the corresponding kernel ($\mathcal{R}$) and use Prop. 14 to compute generators for the corresponding image ($\mathcal{S}$). The result now follows. ◀

The remainder of this section will be devoted to the proof of Prop. 14. Let $\Delta(X)$ be the maximum degree of X. Let $A \subseteq 2^{V(X_r)} \setminus \{\emptyset\}$ contain those subsets of size at most Δ. Define $f : V(X_{r+1}) \setminus V(X_r) \to A$ by: $f(v) := \{w \in V(X_r) : vw \in E(X)\}$. Luks [40] previously established that $\sigma \in \mathrm{Aut}_e(X_r)$ is in $\mathrm{Im}(\pi_r)$ if and only if, for each $0 \leq s \leq m-1$, σ stabilizes the set of fathers of s-tuples: $A_s := \{a \in A : |f^{-1}(a)| = s\}$, as well as the set A' of new edges. Color A accordingly with $2m$ colors (see e.g., [40,

p. 49]). We need to now find the automorphisms in $G = \mathrm{Aut}_e(X_r)$ acting on A, that preserve the edge colors. We now recall some notions from Luks [40]. Let A be a colored set, with coloring χ. Let $B \subseteq A$ and $K \leq \mathrm{Sym}(A)$. The set of permutations in K that preserve the colors in B is: $\mathcal{C}_B(K) := \{\sigma \in K : \text{for all } b \in B, \chi(\sigma(b)) = \chi(b)\}$. Observe that: $\mathcal{C}_B(K \cup K') = \mathcal{C}_B(K) \cup \mathcal{C}_B(K')$, and $\mathcal{C}_{B \cup B'}(K) = \mathcal{C}_B(K) \cap \mathcal{C}_{B'}(K)$. Note that if $\mathcal{C}_B(\sigma B)$ is non-empty, then it is a left-coset of the subgroup $\mathcal{C}_B(G) \leq K$ [40, Lem. 2.4]. We now turn to solving the Color Automorphism problem. For $k \in \mathbb{N}$, let Γ_k be the class of groups G where all the non-Abelian composition factors of G are subgroups of $\mathrm{Sym}(k)$.

The Color Automorphism takes as input generators for a subgroup $G \leq \mathrm{Sym}(A)$ with $G \in \Gamma_k$, a G-stable subset B, and $\sigma \in \mathrm{Sym}(A)$. The solution is $\mathcal{C}_B(\sigma G)$. We will consider the Color Automorphism problem in the case when $k = m \in O(\log n)$. In order to solve the Color Automorphism problem, we proceed via a divide-and-conquer procedure. We have the following cases in the recursion:

Case 1 (Base Case). The recursion bottoms out when $|B| = 1$. In this case, we compute the Pointwise Stabilizer of $\mathcal{C}_{B^{\sigma^{-1}}}(G)$ in $\mathsf{FOLL}^{O(1)}$using Lem. 10(d). Now $\sigma\mathcal{C}_{B^{\sigma^{-1}}}(G) = \mathcal{C}_B(\sigma G)$.

Case 2 (Intransitive Case). Suppose that B is the disjoint union of G-stable subsets $Z_1, \ldots, Z_k$. We use Lem. 10(g) to, in $\mathsf{FOLL}^{O(1)}$, break B up into orbits $Z_1, \ldots, Z_k$. We now have that: $\mathcal{C}_B(\sigma G) = \bigcap_{i=1}^{k} \mathcal{C}_{Z_i}(\sigma G)$. We may thus compute each $\mathcal{C}_{Z_i}(\sigma G)$ in parallel. As G acts transitively on each Z_i, the recursive calls for $\mathcal{C}_{Z_i}(\sigma G)$ fall under Cases 1 or 3. In order to compute the intersection, we use a binary tree circuit of depth $O(\log k) \leq O(\log |X|) \leq O(\log \log n)$. At each node of the binary tree, we utilize the $\mathsf{FOLL}^{O(1)}$ algorithm for CosetInt (Lem. 10h). Thus, the total non-recursive work in this case is $\mathsf{FOLL}^{O(1)}$.

Case 3 (Transitive Case). If $|B| > 1$ and B is not the disjoint union of at least two G-stable subsets, then we find a minimal G-block system in B, which we call Φ. As $|X| \in O(\log n)$, such a block system is $\mathsf{FOLL}^{O(1)}$-computable (Lem. 10(f)). Furthermore, by Lem. 10(c), we may compute the kernel K of the G-action on Φ in $\mathsf{FOLL}^{O(1)}$. From [2], we have that $[G : K] \in (\log n)^{O(\log \log n)}$. Thus, we may write G as a union of $(\log n)^{O(\log \log n)}$ cosets: $G = \bigcup_{i=1}^{[G:K]} \tau_i K$, using an AC circuit of depth $O((\log n)\mathrm{poly}(\log \log n))$ and size $\mathrm{poly}(n)$ (Lem. 11). Now the problem breaks up as follows: $\mathcal{C}_B(\sigma G) = \bigcup_{i=1}^{[G:K]} \mathcal{C}_B(\sigma \tau_i K)$.

In parallel, we recursively compute $\mathcal{C}_B(\sigma \tau_i K)$ for each $1 \leq i \leq [G : K]$. As K fixes (setwise) each block of Φ, the computation for each $\mathcal{C}_B(\sigma \tau_i K)$ falls under Case 1 or Case 2. We may, in parallel, recursively compute each $\mathcal{C}_B(\sigma \tau_i K)$. Each recursive call deals with subsets of size $|B|/\ell$. When we recombine the cosets, we must take care to ensure that we only have $\mathrm{poly}(n)$ generators. We accomplish this in $\mathsf{FOLL}^{O(1)}$using Lem. 10(e). We use a binary tree circuit of depth $\log[G : K] \in O((\log \log n)^2)$ to compute $\mathcal{C}_B(\sigma G)$ from the $\mathcal{C}_B(\sigma \tau_i K)$ $(1 \leq i \leq [G : K])$. This yields a circuit of depth $\mathrm{poly}(\log \log n)$ and size $\mathrm{poly}(n)$ to compute $\mathcal{C}_B(\sigma G)$.

Complexity Analysis. A recursive call at the intransitive case (Case 2) results in either the base case (Case 1) or the transitive case (Case 3). Similarly, a recursive call at the transitive case (Case 3) results in either the base case (Case 1) or the intransitive case (Case 3). The recursive calls in the transitive case reduce the size of the problem by at least 1/2. Thus, the recursion tree thus has height $O(\log m) = O(\log \log n)$. Each level of the recursion tree is computable using an AC circuit of depth $O((\log n)\mathrm{poly}(\log \log n))$ and size $\mathrm{poly}(n)$. Thus, in total, we require an AC circuit of depth $O((\log n)\mathrm{poly}(\log \log n))$ and size $\mathrm{poly}(n)$, as desired. This completes the proof of Prop. 14.

4 Linear Code Equivalence for Small Codes in Parallel

In this section, we establish Thm. 2.

Proof Sketch of Thm. 2. Babai [6] exhibited an algorithm that tests the equivalence of two linear codes of length m in time $(2+o(1))^m$. He accomplished this by reducing to $\binom{m}{d}$ instances of Graph Isomorphism – precisely, isomorphism testing of $d \times (m-d)$ bipartite graphs with colored edges. A careful analysis shows that Babai's reduction is computable using an NC circuit of depth $O(\log^2 m)$ and size $\mathrm{poly}(m)$. As $m \in O(\log n)$, we obtain that this reduction is $\mathsf{FOLL}^{O(1)}$-computable. By Thm. 13, we can decide isomorphism of such graphs using an AC circuit of depth $O((\log^2 n) \cdot \mathrm{poly}(\log\log n))$ and size $\mathrm{poly}(n)$. The result now follows. ◀

5 Isomorphism Testing of Coprime Extensions in Parallel

In this section, we will establish Thm. 1. We begin by recalling some additional preliminaries concerning coprime extensions. We note that coprime extensions are determined entirely by the isomorphism types of N, H and their actions (Lem. 8). We will now show how to compute a decomposition $G = H \ltimes N$, where N is Abelian, H is elementary Abelian, and $\gcd(|H|, |N|) = 1$, if such a decomposition exists.

▶ **Lemma 15.** *Let G be a group given by its multiplication table. We can, in FL, decide if there exists an Abelian normal Hall subgroup $N \trianglelefteq G$ such that G/N is elementary Abelian and $\gcd(|N|, |G/N|) = 1$, as well as compute such an N, if one exists, and G/N in FL.*

Let p, q be distinct primes. We recall key facts regarding representations of $\mathbb{Z}_q^\ell$ over $\mathbb{Z}_p$, as well as additional background from [46, Section 5.2]. For $n \in \mathbb{N}$, the *cyclotomic polynomial* $\Phi_n(x)$ is the unique irreducible polynomial that is a divisor of $x^n - 1$, but for all $k < n$, is not a divisor of $x^k - 1$. Suppose that $\Phi_q(x)$ factors as $g_1(x) \cdots g_r(x)$, where $g_1(x), \ldots, g_r(x) \in \mathbb{Z}_p[x]$ are monic polynomials of degree $d = (q-1)/r$. Note that $d = |\mathbb{Z}_q^\times|$. Let $M \in \mathrm{GL}_d(\mathbb{Z}_p)$ be the companion matrix of $g_1(x)$. For $v \in \mathbb{Z}_q^\ell$ with $v \neq 0$, define $v^* : \mathbb{Z}_q^\ell \to \mathbb{Z}_q$ by mapping $v^*(u) = (v, u)$ (the inner product of v and u).

For $u \in \mathbb{Z}_q^\ell$, let $0 \leq h_u < q$ such that $h_u \equiv v^*(u) \pmod q$. Now define $f_v : \mathbb{Z}_q^\ell \to \mathrm{GL}_d(\mathbb{Z}_p)$ by sending $u \mapsto M^{h_u}$. We write $M^{v^*(u)}$ in place of M^{h_u}. Let $f_0 : \mathbb{Z}_q^\ell \to \mathrm{GL}_d(\mathbb{Z}_p)$ be the trivial representation. Now $\{f_v : v \in \mathbb{Z}_q^\ell\}$ forms the set of all irreducible representations of $\mathbb{Z}_q^\ell$ over $\mathbb{Z}_p$. Note that for distinct and non-zero $u, v \in \mathbb{Z}_q^\ell$, f_u and f_v might be equivalent.

▶ **Lemma 16** ([46, Claim 1])**.** *Let $u, v \in \mathbb{Z}_q^\ell$ be distinct and both non-zero. Let f_u, f_v be the corresponding irreducible representations. We have that f_u and f_v are equivalent if and only if there exists some $s \in \mathbb{Z}_q$ such that $u = sv$, and M^s and M are conjugate (by abuse of notation, we associate s with the least non-negative integer belonging to the equivalence class $s \in \mathbb{Z}_q$).*

Let $\tau : \mathbb{Z}_q^\ell \to \mathrm{GL}_k(\mathbb{Z}_p)$. As p, q are distinct primes, we have by Maschke's theorem that τ is completely reducible. Write $\tau = f_{v_1}^{k_1} \oplus \cdots \oplus f_{v_t}^{k_t}$, where each $v_i \in \mathbb{Z}_q^\ell$, and $k_1 \geq k_2 \geq \cdots \geq k_t \geq 1$. For a given multiplicity $w \in [k]$, define $L_\tau(w)$ to be the set of irreducible representations with multiplicity w appearing in τ. Now define $L_\tau := (L_\tau(w))_{w \in [k]}$. We have that L_τ determines τ up to equivalence. In order to deal with the concrete form of the representations, Qiao, Sarma, and Tang introduced the following.

▶ **Definition 17** ([46, Definition 5.4]). *Let $\tau : \mathbb{Z}_q^\ell \to GL_k(\mathbb{Z}_p)$, and let $w \in [k]$. Define $\mathcal{L}_\tau(w)$ to be a set of vectors such that for every irreducible representation $f \in L_\tau(w)$, there exists a unique $v \in \mathcal{L}_\tau(w)$ such that f and f_v are equivalent. Now define $\mathcal{L}_\tau := (\mathcal{L}_\tau(w))_{w\in[k]}$. We refer to such a tuple $\mathcal{L}_\tau$ as an* indexing tuple *of L_τ.*

A representation $\tau : \mathbb{Z}_q^\ell \to \mathrm{GL}_k(\mathbb{Z}_p)$ has at most 2^k indexing tuples [46].

▶ **Proposition 18** ([46, Claim 2]). *Let $\tau, \gamma : \mathbb{Z}_q^\ell \to GL_k(\mathbb{Z}_p)$ be two representations. We have that τ and γ are equivalent if and only if there exist indexing tuples of τ and γ, $\mathcal{L}_\tau$ and $\mathcal{L}_\gamma$, such that $\mathcal{L}_\tau = \mathcal{L}_\gamma$.*

▶ **Lemma 19.** *Let p, q be distinct primes, and let $G = \mathbb{Z}_q^\ell \ltimes_\varphi \mathbb{Z}_p^k$ be given by its multiplication table. We can list all indexing tuples of φ in NC^3.*

▶ **Theorem 20** (cf. [46, Thm. 1.3]). *Given groups G_1, G_2 by their multiplication tables, there exists a uniform AC^3 algorithm that decides if $G_1, G_2 \in \mathcal{H}(\mathcal{E}, \mathcal{E})$; and if so, decides if $G_1 \cong G_2$.*

Proof. We first use Lem. 15 to, in FL, find decompose $G_i = H_i \ltimes_{\tau_i} N_i$ $(i = 1, 2)$. We may then, in L, decide whether: (i) the N_i and H_i are elementary Abelian groups of coprime order, (ii) $N_1 \cong N_2$, and (iii) $H_1 \cong H_2$ [11]. Suppose that $N_1 \cong N_2 \cong \mathbb{Z}_p^k$ and $H_1 \cong H_2 \cong \mathbb{Z}_q^\ell$. It remains to decide whether $\tau_1, \tau_2 : \mathbb{Z}_q^\ell \to \mathrm{GL}_k(\mathbb{Z}_p)$ are equivalent. By Prop. 12, we may in AC^1, decompose τ_1, τ_2 into their irreducible components and group them by equivalence type. Write $\tau_1 = f_{v_1}^{k_1} \oplus \cdots \oplus f_{v_t}^{k_t}$ and $\tau_2 = f_{u_1}^{\ell_1} \oplus \cdots \oplus f_{u_{t'}}^{\ell_{t'}}$. Now in NC^3 (using Lem. 19), we may obtain indexing tuples $\mathcal{L}_{\tau_1}, \mathcal{L}_{\tau_2}$. We may check in L that $t = t'$; as well as that for all $w \in [k]$, whether $|\mathcal{L}_{\tau_1}(w)| = |\mathcal{L}_{\tau_2}(w)|$. If these conditions are not satisfied, then τ_1 and τ_2 are not equivalent; in which case, G_1 and G_2 are not isomorphic.

We now consider our fixed indexing tuple $\mathcal{L}_{\tau_1}$ for τ_1. By Prop. 18, it suffices to decide if there exists an indexing tuple $\mathcal{L}_{\tau_2}$ of τ_2 and $\varphi \in \mathrm{GL}_\ell(\mathbb{Z}_p)$ such that $\mathcal{L}_{\tau_1}^\varphi = \mathcal{L}_{\tau_2}$. In parallel, we will consider all indexing tuples of τ_2. Note that we can list all such indexing tuples of τ_2 in NC^3 (using Lem. 19). For clarity, fix such an indexing tuple $\mathcal{L}_{\tau_2}$. Deciding whether such a φ exists reduces to CodeEq for a code of length $m \in O(\log n)$ followed by taking an intersection of the code equivalences with $\prod_{w=1}^k \mathrm{Sym}(|\mathcal{L}_{\tau_1}(w)|)$ [46, Prop. 5]. We can write down standard 2-element generating sets for $\mathrm{Sym}(|\mathcal{L}_{\tau_1}(w)|)$ in AC^0. By Thm. 2, we can solve this instance of CodeEq using an AC circuit of depth $O((\log^2 n) \cdot \mathrm{poly}(\log\log n))$ and size $\mathrm{poly}(n)$. The requisite instance of CosetInt is $\mathsf{FOLL}^{O(1)}$-computable (Lem. 10(h)). ◀

In order to handle the case of $\mathcal{H}(\mathcal{A}, \mathcal{E})$, we use the following to reduce to the case of $\mathcal{H}(\mathcal{E}, \mathcal{E})$.

▶ **Proposition 21** (cf. [36, 8]). *Let A be an Abelian p-group given by its multiplication table. Let $S = (g_1, \ldots, g_s)$ be a basis for A. Let $\varphi_1, \varphi_2 \in Aut(A)$ be given as matrices with respect to S. Suppose that $|\varphi_1| = |\varphi_2|$. If p does not divide $|\varphi_1| = |\varphi_2|$, then there exists an FL-computable map $\Psi_p : Aut(A) \to GL_s(\mathbb{Z}_p)$ such that φ_1 and φ_2 are conjugate if and only if $\Psi_p(\varphi_1)$ and $\Psi_p(\varphi_2)$ are conjugate.*

Following the strategy of [36, 46, 8], we will later use Prop. 21 to reduce isomorphism testing of $\mathcal{H}(\mathcal{A}, \mathcal{E})$ to the case of $\mathcal{H}(\mathcal{E}, \mathcal{E})$. Le Gall [36] established Prop. 21 in the case when the complement was cyclic. He also showed that this reduction is NC-computable relative to $|A|$. Babai and Qiao [8] subsequently showed that Le Gall's reduction holds for arbitrary complements. In the full version, we will carefully analyze this reduction, to show that Ψ_p is FL-computable. This is a critical step in order to establish the AC^3 bounds in Thm. 1. We will now prove Thm. 1.

Proof of Thm. 1. Using Lem. 15, we write $G_i = H_i \ltimes_{\varphi_i} N_i$. Now in L, we test whether H_i is elementary Abelian, $N_1 \cong N_2$, and $H_1 \cong H_2$ [11, 19]. Otherwise, we reject. For $i \in [2]$, let $A_{i,p} \leq N_i$ be the Sylow p-subgroup of N_i (and hence, G_i). Let $\varphi_{i,p} : H_i \to \mathrm{Aut}(A_{i,p})$ be the projection of φ_i to $A_{i,p}$. Let $G_{i,p} = H_i \ltimes \varphi_{i,p} A_{i,p}$. By [46, Section 5.3], we have that $G_1 \cong G_2$ if and only if, for all primes p dividing $|N_1| = |N_2|$, $G_{1,p} := H_1 \ltimes_{\varphi_{1,p}} A_{1,p}$ and $G_{2,p} := H_2 \ltimes_{\varphi_{2,p}} A_{2,p}$ are isomorphic. We have shown that, in FL, we can construct each $G_{1,p}, G_{2,p}$. By [33, Prop. 4.5], we may in AC^1 compute a basis β for $A_{1,p}$, as well as $A_{2,p}$, and thus fix an isomorphism $\phi_p : A_{1,p} \cong A_{2,p}$. Let Ψ_p be as defined in Prop. 21. By Prop. 21 (applied with β), we have that $\varphi_{1,p}, \varphi_{2,p}$ are conjugate if and only if $\Psi_p(\varphi_{1,p})$ and $\Psi_p(\varphi_{2,p})$ are conjugate. Furthermore, Ψ_p is FL-computable. Thus, given $G_{1,p}$ and $G_{2,p}$, we may in AC^1 write down $G'_{i,p} := H_1 \ltimes_{\Psi_p(\varphi_{i,p})} \mathbb{Z}_p^{s_p}$ $(i = 1, 2)$. As $H_1 \cong H_2$ and $A_{1,p} \cong A_{2,p}$, we have that the following are equivalent: (i) $G_{1,p} \cong G_{2,p}$, (ii) $G'_{1,p} \cong G'_{2,p}$, and (iii) $\Psi_p(\varphi_1)$ and $\Psi_p(\varphi_2)$ are conjugate. Thus, it suffices to test for isomorphism between $G'_{1,p} \cong G'_{2,p}$, which is AC^3-computable using Thm. 20. The result now follows. ◀

6 Isomorphism Testing of Central-Radical Groups in Parallel

6.1 When Enumerating Aut(Q) is Allowed

In this section, we will establish the following.

▶ **Theorem 22** (cf. [30, Thm. 6.1]). *Let $\mathcal{S}$ be a logspace-computable characteristic subgroup function. Fix functions $d(n) \in \Omega(\log^2 n)$ and $s(n)$. Let G_1, G_2 be two groups of order n given by their multiplication tables, and suppose that (i) $\mathcal{S}(G_1) \leq Z(G_1)$ and (ii) $Aut(G_1/\mathcal{S}(G_1))$ can be listed using an AC circuit of depth $d(n)$ and size $s(n)$. Then we can decide isomorphism between G_1 and G_2 using an AC circuit of depth $\max\{d(n), \log^3(n)\}$ and size $s(n) \cdot n^{O(1)}$.*

Note that $\mathrm{Aut}(G/\mathrm{Rad}(G))$ can be listed using an an AC circuit of depth $O(\log \log n)$ and size $n^{O(\log \log n)}$ [26], which yields:

▶ **Corollary 23** (cf. [30, Thm. A]). *Isomorphism of central-radical groups, given by their multiplication tables, can be decided by AC circuits of depth $O(\log^3 n)$ and size $n^{O(\log \log n)}$.*

We recall from [30] details on how to work with 2-cohomology classes algorithmically. First, as the action is trivial in central extensions, we will drop it from $Z^2(Q, A), B^2(Q, A)$, and $H^2(Q, A)$. Write $A := \mathcal{S}(G)$, and let $A = \prod_{i=1}^{k} \mathbb{Z}/p_i^{\mu_i}\mathbb{Z}$ be the decomposition of A into cyclic subgroups. By choosing an arbitrary section s, we get a cocycle $f : Q \times Q \to A$. We may view f as a $k \times |Q|^2$-size integer matrix, which we denote M_f. The rows are indexed by $[k]$, and the columns are indexed by $Q \times Q$. For $i \in [k]$ and $(q, q') \in Q \times Q$, the entry $M_f[i, (q, q')]$ is the ith coordinate of $f(q, q')$ relative to the basis $(e_1, \ldots, e_k)$ modulo $p_i^{\mu_i}$.

Under this identification, the set $C^2(Q, A)$ is the set of all such matrices. Now $Z^2(Q, A)$ is a subgroup of $C^2(Q, A)$, under matrix addition. Similarly, $B^2(Q, A)$ is a subgroup of $Z^2(Q, A)$ under matrix addition. We will use U_i to denote the subgroup of $C^2(Q, A)$ consisting of matrices whose only non-zero entries are in the ith row (so $U_i \cong (\mathbb{Z}/p_i^{\mu_i}\mathbb{Z})^{|Q|^2}$). $\mathrm{Aut}(A)$ acts on $C^2(Q, A)$ by left-multiplication, and $\mathrm{Aut}(Q)$ acts on $C^2(Q, A)$ by permuting the columns according to the diagonal action of $\mathrm{Aut}(Q)$. Note that the actions of $\mathrm{Aut}(A)$ and $\mathrm{Aut}(Q)$ commute.

▶ **Proposition 24** (cf. [30, Prop. 6.8]). *For any $\mu \geq 1$, a $\mathbb{Z}$-basis of $B^2(Q, \mathbb{Z}/p^\mu\mathbb{Z})$ can be computed in AC^0. Furthermore, a $\mathbb{Z}_p$-basis of $B^2(Q, \mathbb{Z}_p)$ can be computed in the same bound.*

Now for a 2-cochain $f \in C^2(Q, A)$ with corresponding $k \times |Q|^2$ matrix M, let $R_i \leq (\mathbb{Z}/p_i^{\mu_i}\mathbb{Z})^{|Q|^2}$ be the subgroup generated by the ith row of M. For $\mu < \mu_i$, let $R_i^{(\mu)}$ denote the subgroup of $(\mathbb{Z}/p_i^{\mu_i}\mathbb{Z})^{|Q|^2}$ that is given by taking R_i modulo p^μ. For $\mu > \mu_i$, let $R_i^{(\mu)}$ denote the subgroup of $(\mathbb{Z}/p_i^{\mu_i}\mathbb{Z})^{|Q|^2}$ that is given by multiplying every element of R_i by $p^{\mu-\mu_i}$. For any prime $q \neq p$, let $R_i^{(q,\mu)}$ denote the trivial subgroup. If $q = p$, let $R_i^{(q,\mu)} := R_i^{(\mu)}$. Let $R^{(p,\mu)} = \langle R_1^{(p,\mu)}, \ldots, R_k^{(p,\mu)} \rangle$.

▶ **Proposition 25** ([30, Prop. 6.9]). *Let $A = \prod_{i=1}^k \mathbb{Z}/p_i^{\mu_i}\mathbb{Z}$ be an Abelian group (the p_i are primes, not necessarily distinct). Let $f_1, f_2 \in C^2(Q, A)$. With the notation as above, there exists $\alpha \in Aut(A)$ such that f_1 and f_2^α are cohomologous if and only if $\langle R_1^{(p_i,\mu_i)}, B^2(Q, \mathbb{Z}/p_i^{\mu_i}\mathbb{Z}) \rangle = \langle R_2^{(p_i,\mu_i)}, B^2(Q, \mathbb{Z}/p_i^{\mu_i}\mathbb{Z}) \rangle$, for each $i \in [k]$. Here, $\langle \cdot \rangle$ denotes the $\mathbb{Z}$-span (=group generated by).*

Proof of Thm. 22. We follow the strategy of [30, Thm. 6.1]. We begin by listing $\mathrm{Aut}(Q)$ using an AC circuit of size $s(n)$ and depth $d(n)$. Choose arbitrary sections of G, H, to get a 2-cocycle f_1 of G and a 2-cocycle f_2 of H. By Lem. 7, it is sufficient and necessary to test whether there exist $(\alpha, \beta) \in \mathrm{Aut}(A) \times \mathrm{Aut}(Q)$ such that f_1 and $f_2^{(\alpha,\beta)}$ are cohomologous.

For each $\beta \in \mathrm{Aut}(Q)$, we get $f_2' := f_2^{(\mathrm{id},\beta)}$. We first use Prop. 24 to, in AC^0, get a basis V for $B^2(Q, A)$. Let M_1 be the matrix representation of f_1, and M_2 be the matrix representation for f_2'. By Prop. 25, it suffices to determine whether the $\mathbb{Z}$-span of the rows in M_1 with V, is the same as the $\mathbb{Z}$-linear span of the rows of M_2 with V. Checking this condition reduces to solving a system of linear equations over the Abelian group A, which is NC^3-computable [42, 45]. The result now follows. ◀

6.2 When $\mathrm{Aut}(Q)$ is too big

We will establish Thm. 3. We first recall additional preliminaries from [30, Sec. 7]. Throughout this section, we will consider central extensions $A \hookrightarrow G_j \twoheadrightarrow Q$ $(j = 1, 2)$, with $A = Z(G_j)$, and $Q = \prod_{i=1}^\ell T_i$ with each T_i perfect, centerless, and indecomposable.

▶ **Proposition 26** (cf. [30, Prop. 6.10]). *For $A = \mathbb{Z}_p^k$ and a group Q, let $n = |A| \cdot |Q|$. In NC^2, we can compute an $Aut(A)$-invariant complement W of $B^2(Q, A)$ in $C^2(Q, A)$, as well as a $\mathbb{Z}_p$-linear projection $\pi : C^2(Q, A) \to W$ that commutes with every $\alpha \in Aut(A)$.*

The following lemma from [30] establishes conditions in which the *cohomology splits*, allowing us to restrict attention to the extensions corresponding to the individual direct factors of the quotient.

▶ **Lemma 27** ([30, Lem. 7.4]). *Given two central extensions $A \hookrightarrow G_j \twoheadrightarrow Q$ $(j = 1, 2)$, with $A = Z(G_j)$, and $Q = \prod_{i=1}^\ell T_i$ with each T_i perfect, centerless, and indecomposable. Let $U_{j,i}$ be the preimage of T_i under the natural projection map $G_j \to G_j/A$. The extensions $A \hookrightarrow G_j \twoheadrightarrow Q$ $(j = 1, 2)$ are equivalent if and only if for all $i \in [\ell]$, the extensions $A \hookrightarrow U_{j,i} \twoheadrightarrow T_i$ $(j = 1, 2)$ are equivalent.*

▶ **Lemma 28.** *Let G be a group of order n. We can decide if (i) $Rad(G) = Z(G)$ is elementary Abelian; and if so, (ii) compute the decomposition of $G/Z(G)$ into a direct product of either non-Abelian simple groups or $O(1)$-size perfect, indecomposable groups in FL. Furthermore, we can group the direct factors by isomorphism type in FL.*

Let $Q = \prod_{i=1}^\ell T_i$. Classify the T_i's together and group them by their isomorphism types, identifying $Q = \prod_{i=1}^r Q_i^{\ell_i}$. Then $\mathrm{Aut}(Q) \cong \prod_{i=1}^r \mathrm{Aut}(Q_i) \wr \mathrm{Sym}(\ell_i)$. A *diagonal* of $\mathrm{Aut}(Q)$ is an element in $\prod_{i=1}^r \mathrm{Aut}(Q_i)$. We can efficiently enumerate the diagonals:

▶ **Lemma 29.** *Given the decomposition $Q = \prod_{i=1}^{r} T_i^{\ell}$ as in the preceding paragraph, we can enumerate all diagonals of Q in* FL.

▶ **Lemma 30** ([30, Lem. 7.5])**.** *Let $A' \times A'' \overset{\iota}{\hookrightarrow} G \overset{\pi}{\twoheadrightarrow} Q$ be a central extension of $A' \times A''$ by Q. Let $p_{A'} : A' \times A'' \to A'$ be the projection onto A' along A''. If there is a 2-cocycle $f : Q \times Q \to A' \times A''$ such that $p_{A'} \circ f$ is a 2-coboundary, then G is isomorphic (even equivalent) to $A' \times (G/A')$. Furthermore, if $Z(G)$ is elementary Abelian, then A' can be computed in* NC^2 *using linear algebra over Abelian groups.*

We recall some notions from [30, Proof of Thm. 7.1]. Let G be a group under consideration in Thm. 3. We say that a 2-cocycle $f_j : Q \times Q \to A$ respects the direct factors if there exist $f_{j,i} : T_i \times T_i \to A$ $(i \in [\ell])$ such that the following condition holds: $f_j((p_1, \dots, p_\ell), (q_1, \dots, q_\ell)) = \sum_{i \in [\ell]} f_{j,i}(p_i, q_i)$. Let $Z^2_{\text{prod}}(Q, A)$ denote the set of 2-cocycles respecting the direct factors. Grochow and Qiao [30, Lem. 7.4] showed that $Z^2_{\text{prod}}(Q, A) \neq \emptyset$. Similarly, we say that a 2-coboundary $b_j : Q \times Q \to A$ respects the direct factors if there exist $b_{j,i} : T_i \times T_i \to A$ $(i \in [\ell])$ such that the following condition holds: $b_j((p_1, \dots, p_\ell), (q_1, \dots, q_\ell)) = \sum_{i \in [\ell]} b_{j,i}(p_i, q_i)$.

Let $B^2_{\text{prod}}(Q, A)$ be the set of 2-coboundaries respecting the direct factors. The difference of two cohomologous 2-cocycles in $Z^2_{\text{prod}}(Q, A)$ belongs to $B^2_{\text{prod}}(Q, A)$. Define $C^2_{\text{prod}}(Q, A)$ as the set of 2-cochains respecting the direct factors. We may view the elements of $C^2_{\text{prod}}(Q, A)$ as $k \times \left(\sum_{i \in [\ell]} |T_i|^2\right)$ matrices, whose rows are indexed by $[k]$ and whose columns are indexed by triples $(i; p, q)$ with $p, q \in T_i$. That is, $C^2_{\text{prod}}(Q, A) = \bigoplus_{i \in [k]} C^2(T_i, A)$.

Proof of Thm. 3. We follow the strategy of [30, Sec. 7.3]. We decompose G_j $(j = 1, 2)$ as an extension of $A = \text{Rad}(G_i) = \mathbb{Z}_p^k$ by $Q = \prod_{i=1}^{\ell} T_i$. Fix arbitrary sections $s_j : Q \to G_j$ $(j = 1, 2)$, and let f_j be the corresponding 2-cocycles. We then classify the T_i's according to their isomorphism types. Grouping them together by isomorphism type, we have $Q = \prod_{i=1}^{r} Q_i^{\ell_i}$. By Lem. 28, this step is FL-computable. We enumerate all diagonals of $\text{Aut}(Q)$ in FL using Lem. 29. By Lem. 7, $G_1 \cong G_2$ if and only if they are pseudo-congruent extensions of A by Q. The extensions are pseudo-congruent if and only if there exists $(\alpha, \beta) \in \text{Aut}(A) \times \text{Aut}(Q)$ such that, after twisting by (α, β), the resulting extensions are equivalent. Once we fix such an $(\alpha, \beta) \in \text{Aut}(A) \times \text{Aut}(Q)$, Lem. 27 provides that the problem is reduced to determining the equivalence of $G_1|_{T_i}$ and $G_2|_{T_i}$ $(i \in [\ell])$.

(a) Consider the case when each T_i is non-Abelian simple. In this case, each T_i is 2-generated, we can list $\text{Aut}(T_i)$ in FL [51]. By Thm. 22, we can determine the equivalence type of $G|_{T_i}$ in AC^3. Note that every $\beta \in \text{Aut}(Q)$ can be represented as a pair $(\delta, \sigma) \in (\prod_{i=1}^{r} \text{Aut}(Q_i)^{\ell_i}) \rtimes (\prod_{i=1}^{r} \text{Sym}(\ell_i))$. Thus, $Z^2_{\text{prod}}(Q, A)$ is an invariant subset of $Z^2(Q, A)$ under the actions of both $\text{Aut}(A)$ and $\text{Aut}(Q)$. Similarly, $B^2_{\text{prod}}(Q, A)$ is an invariant subset of $B^2(Q, A)$ under the actions of both $\text{Aut}(A)$ and $\text{Aut}(Q)$. By Prop. 26, we can in NC^2, compute for each $i \in [\ell]$ an $\text{Aut}(A)$-invariant W_i such that $C^2(T_i, A) = B^2(T_i, A) \oplus W_i$, along with a projection $\pi_i : C^2(T_i, A) \to W_i$ such that π_i commutes with the action of $\text{Aut}(A)$.
From [30, Sec. 7.3], we have the following properties. Each element of W_i $(i \in [\ell])$ can be written as a $k \times 17$ matrix, in a manner that is still $\text{Aut}(A)$-equivariant. For the remainder of this proof, we will denote π_i as the composition of the previous π_i, followed by the mapping onto $k \times 17$ matrices. Furthermore, for each choice of diagonal $\delta = \prod_{i=1}^{\ell} \text{Aut}(T_i)$, we may choose the complements W_i, W_h such that whenever $T_i \cong T_h$, δ identifies W_i and W_h. With this choice, we can construct $\pi_\delta = \bigoplus_{i=1}^{\ell} \pi_i : C^2(Q, A) \to W \leq \text{Mat}_{k \times 17\ell}(\mathbb{Z}_p)$, for some $\text{Aut}(A)$-invariant $W \leq C^2_{\text{prod}}(Q, A)$ such that $C^2_{\text{prod}}(Q, A) = B^2_{\text{prod}}(Q, A) \oplus W$ and π_δ commutes with the action of $\text{Aut}(A) \times \prod_{i=1}^{r} \text{Sym}(\ell_i)$. For each diagonal δ,

it remains to decide whether there exists $(\alpha, \sigma) \in \mathrm{Aut}(A) \times \prod_{i=1}^{r} \mathrm{Sym}(\ell_i)$ such that $\pi(f_1) = \pi(f_2^{(\mathrm{id},\delta,\mathrm{id})})^{(\alpha,\mathrm{id},\sigma)}$. Note that as α, δ commute and σ is already on the right, we have that $(\alpha, \delta, \sigma) = (\mathrm{id}, \delta, \mathrm{id})(\alpha, \mathrm{id}, \sigma)$.
Let $M_1 := M_{\pi(f_1)}$. Without loss of generality, we may assume that M_1 has rank k. Otherwise, we may in NC^2 (using Lem. 30), split out a direct factor of the center as $A' \times G_j/A'$ $(j = 1, 2)$. By the Remak–Krull–Schmidt theorem, we then reduce to testing isomorphism between G_1/A' and G_2/A', where the desired rank condition holds. Now in parallel, for each diagonal δ of $\prod_{i=1}^{\ell} T_i$, we compute $\pi(f_1)$ and $\pi(f_2^{(\mathrm{id},\delta,\mathrm{id})})$. Let $M_1 = M_{\pi(f_1)}$ and $M_2 = M_{\pi(f_2^{(\mathrm{id},\delta,\mathrm{id})})}$. Again note that we can enumerate all such diagonals in FL (Lem. 29). Furthermore, we have established above that π can be constructed in NC^2.
As the action of $\mathrm{Aut}(A)$ on M_1, M_2 is by left multiplication, and the action of $\prod_{i=1}^{r} \mathrm{Sym}(\ell_i)$ is on the blocks of columns, we treat M_1, M_2 as generators of two $\mathbb{Z}_p$-codes of dimension k and length 17ℓ. As $\ell \leq \log n$, we compute the coset of equivalences $\mathrm{CodeEq}(M_1, M_2) \subseteq \mathrm{Sym}(17\ell)$ using an AC circuit of depth $O((\log^2 n)(\mathrm{poly}(\log \log n)))$ and size $\mathrm{poly}(n)$ (Thm. 2). We then compute $\mathrm{CodeEq}(M_1, M_2) \cap \prod_{i=1}^{r} \mathrm{Sym}(\ell_i)$ in $\mathsf{FOLL}^{O(1)}$using Lem. 10h. If this intersection is non-empty, then we report isomorphic. Using a single OR gate, we return true (report isomorphic) if and only if at least one of these intersections is non-empty (where the OR is taken over all diagonals of Q). The total work is computable in AC^3.

(b) Consider now the case in which each T_i has size at most $D \in O(1)$. This case is handled almost identically as in (a). We sketch the differences. In this case, we can determine the isomorphism type of a given T_i in AC^0, by trying all of the $D! \in O(1)$ permutations. The corresponding linear codes have length $\ell \cdot D^2$, rather than 17ℓ [30]. The total work is computable in AC^3. ◀

References

1 V. Arvind and Piyush P. Kurur. Graph isomorphism is in SPP. *Information and Computation*, 204(5):835–852, 2006. doi:10.1016/j.ic.2006.02.002.

2 L Babai, P.J Cameron, and P.P Pálfy. On the orders of primitive groups with restricted nonabelian composition factors. *Journal of Algebra*, 79(1):161–168, 1982. doi:10.1016/0021-8693(82)90323-4.

3 L. Babai, E. Luks, and A. Seress. Permutation groups in NC. In *STOC 1987*, STOC '87, pages 409–420, New York, NY, USA, 1987. Association for Computing Machinery. doi:10.1145/28395.28439.

4 László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *STOC'16—Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 684–697. ACM, New York, 2016. Preprint of full version at arXiv:1512.03547v2 [cs.DS]. doi:10.1145/2897518.2897542.

5 László Babai and Robert Beals. A polynomial-time theory of black box groups I. In *Groups St Andrews 1997 in Bath, I*, 1999.

6 László Babai, Paolo Codenotti, Joshua A. Grochow, and Youming Qiao. Code equivalence and group isomorphism. In *Proceedings of the Twenty-Second Annual ACM–SIAM Symposium on Discrete Algorithms (SODA11)*, pages 1395–1408, Philadelphia, PA, 2011. SIAM. doi:10.1137/1.9781611973082.107.

7 László Babai, Paolo Codenotti, and Youming Qiao. Polynomial-time isomorphism test for groups with no abelian normal subgroups - (extended abstract). In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 51–62, 2012. doi:10.1007/978-3-642-31594-7_5.

8 László Babai and Youming Qiao. Polynomial-time isomorphism test for groups with Abelian Sylow towers. In *29th STACS*, pages 453–464. Springer LNCS 6651, 2012. doi:10.4230/LIPIcs.STACS.2012.453.

9 László Babai and Endre Szemerédi. On the complexity of matrix group problems I. In *25th Annual Symposium on Foundations of Computer Science, West Palm Beach, Florida, USA, 24-26 October 1984*, pages 229–240. IEEE Computer Society, 1984. doi:10.1109/SFCS.1984.715919.

10 D.A.M. Barrington. Quasipolynomial size circuit classes. In *[1992] Proceedings of the Seventh Annual Structure in Complexity Theory Conference*, pages 86–93, 1992. doi:10.1109/SCT.1992.215383.

11 David A. Mix Barrington, Peter Kadau, Klaus-Jörn Lange, and Pierre McKenzie. On the complexity of some problems on groups input as multiplication tables. *J. Comput. Syst. Sci.*, 63(2):186–200, 2001. doi:10.1006/jcss.2001.1764.

12 Hans Ulrich Besche and Bettina Eick. Construction of finite groups. *J. Symb. Comput.*, 27(4):387–404, 1999. doi:10.1006/jsco.1998.0258.

13 Hans Ulrich Besche, Bettina Eick, and E.A. O'Brien. A millennium project: Constructing small groups. *Intern. J. Alg. and Comput*, 12:623–644, 2002. doi:10.1142/S0218196702001115.

14 Jendrik Brachter. *Combinatorial approaches to the group isomorphism problem*. PhD thesis, TU Darmstadt, September 2023. URL: https://tuprints.ulb.tu-darmstadt.de/26387/1/thesis_brachter.pdf.

15 Jendrik Brachter and Pascal Schweitzer. A systematic study of isomorphism invariants of finite groups via the Weisfeiler–Leman dimension, 2022. doi:10.4230/LIPIcs.ESA.2022.27.

16 Peter A. Brooksbank, Joshua A. Grochow, Yinan Li, Youming Qiao, and James B. Wilson. Incorporating Weisfeiler–Leman into algorithms for group isomorphism. arXiv:1905.02518 [cs.CC], 2019.

17 Harry Buhrman and Steven Homer. Superpolynomial circuits, almost sparse oracles and the exponential hierarchy. In R. K. Shyamasundar, editor, *Foundations of Software Technology and Theoretical Computer Science, 12th Conference, New Delhi, India, December 18-20, 1992, Proceedings*, volume 652 of *Lecture Notes in Computer Science*, pages 116–127. Springer, 1992. doi:10.1007/3-540-56287-7_99.

18 John J. Cannon and Derek F. Holt. Automorphism group computation and isomorphism testing in finite groups. *J. Symb. Comput.*, 35:241–267, March 2003. doi:10.1016/S0747-7171(02)00133-5.

19 Arkadev Chattopadhyay, Jacobo Torán, and Fabian Wagner. Graph isomorphism is not AC^0-reducible to group isomorphism. *ACM Trans. Comput. Theory*, 5(4):Art. 13, 13, 2013. Preliminary version appeared in FSTTCS '10; ECCC Tech. Report TR10-117. doi:10.1145/2540088.

20 Nathaniel A. Collins, Joshua A. Grochow, Michael Levet, and Armin Weiß. On the constant-depth circuit complexity of generating quasigroups. *TheoretiCS*, Volume 4, August 2025. Preliminary version in Proceedings of the 2024 International Symposium on Symbolic and Algebraic Computation, ISSAC 2024 (DOI:10.1145/3666000.3669691). doi:10.46298/theoretics.25.19.

21 Heiko Dietrich and James B. Wilson. Group isomorphism is nearly-linear time for most orders. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 457–467, 2022. doi:10.1109/FOCS52979.2021.00053.

22 John D. Dixon and Brian Mortimer. *Permutation groups*, volume 163 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1996. doi:10.1007/978-1-4612-0731-3.

23 Bettina Eick, C. R. Leedham-Green, and E. A. O'Brien. Constructing automorphism groups of p-groups. *Comm. Algebra*, 30(5):2271–2295, 2002. doi:10.1081/AGB-120003468.

24 V. Felsch and J. Neubüser. On a programme for the determination of the automorphism group of a finite group. In *Computational Problems in Abstract Algebra (Proc. Conf., Oxford, 1967)*, pages 59–60. 1970.

25 Flavio Ferrarotti, Senén González, Klaus-Dieter Schewe, and José María Turull-Torres. Proper hierarchies in polylogarithmic time and absence of complete problems. In Andreas Herzig and Juha Kontinen, editors, *Foundations of Information and Knowledge Systems*, pages 90–105, Cham, 2020. Springer International Publishing. doi:10.1007/978-3-030-39951-1_6.

26 Joshua A. Grochow, Dan Johnson, and Michael Levet. Complexity of identifying Fitting-free groups. In Artur Jez and Jan Otop, editors, *Fundamentals of Computation Theory - 25th International Symposium, FCT 2025, Wrocław, Poland, September 15-17, 2025, Proceedings*, volume 16106 of *Lecture Notes in Computer Science*, pages 208–220. Springer, 2025. Preprint of full version at arXiv:2504.19777 [cs.CC]. doi:10.1007/978-3-032-04700-7_16.

27 Joshua A. Grochow and Michael Levet. On the descriptive complexity of groups without abelian normal subgroups. *Logical Methods in Computer Science*, Volume 21, Issue 4, November 2025. Preliminary version appeared in the proceedings of GandALF 2023 (DOI:10.4204/EPTCS.390.12). doi:10.46298/lmcs-21(4:20)2025.

28 Joshua A. Grochow and Michael Levet. On the parallel complexity of group isomorphism via Weisfeiler–Leman. *Journal of Computer and System Sciences*, 156:103703, 2026. Preliminary version appeared in the proceedings of FCT 2023, DOI:10.1007/978-3-031-43587-4_17. doi:10.1016/j.jcss.2025.103703.

29 Joshua A. Grochow and Youming Qiao. Polynomial-time isomorphism test of groups that are tame extensions - (extended abstract). In *Algorithms and Computation - 26th International Symposium, ISAAC 2015, Nagoya, Japan, December 9-11, 2015, Proceedings*, pages 578–589, 2015. doi:10.1007/978-3-662-48971-0_49.

30 Joshua A. Grochow and Youming Qiao. Algorithms for group isomorphism via group extensions and cohomology. *SIAM J. Comput.*, 46(4):1153–1216, 2017. Preliminary version in IEEE Conference on Computational Complexity (CCC) 2014 (DOI:10.1109/CCC.2014.19). Also available as arXiv:1309.1776 [cs.DS] and ECCC Technical Report TR13-123. doi:10.1137/15M1009767.

31 D. Holt, B. Eick, and E. O'Brien. *Handbook of Computational Group Theory*. Chapman and Hall/CRC, 2005.

32 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

33 Dan Johnson, Michael Levet, Petr Vojtěchovský, and Brett Widholm. Parallel complexity of identifying groups and quasigroups via decompositions. In Uli Fahrenberg, Wesley Fussner, and Luigi Santocanale, editors, *Relational and Algebraic Methods in Computer Science*, pages 188–207, Cham, 2026. Springer Nature Switzerland. Preprint of full version at arXiv:2508.06478 [cs.DS]. doi:10.1007/978-3-032-22469-9_11.

34 William M. Kantor, Eugene M. Luks, and Peter D. Mark. Sylow subgroups in parallel. *J. Algorithms*, 31(1):132–195, 1999. doi:10.1006/JAGM.1999.0982.

35 Johannes Köbler, Uwe Schöning, and Jacobo Torán. Graph isomorphism is low for PP. *Comput. Complex.*, 2:301–330, 1992. doi:10.1007/BF01200427.

36 François Le Gall. Efficient isomorphism testing for a class of group extensions. In *Proc. 26th STACS*, pages 625–636, 2009. doi:10.4230/LIPIcs.STACS.2009.1830.

37 François Le Gall and David J. Rosenbaum. On the group and color isomorphism problems. arXiv:1609.08253 [cs.CC], 2016.

38 Michael Levet, Puck Rombach, and Nicholas Sieger. Canonizing Graphs of Bounded Rank-Width in Parallel via Weisfeiler-Leman. In Hans L. Bodlaender, editor, *19th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2024)*, volume 294 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 32:1–32:18, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SWAT.2024.32.

39 Eugene Luks. Permutation groups and polynomial-time computation. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 11, September 1993. doi:10.1090/dimacs/011/11.

40 Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25(1):42–65, 1982. doi:10.1016/0022-0000(82)90009-5.

41 Eugene M. Luks. Group isomorphism with fixed subnormal chains, 2015. arXiv:1511.00151.

42 Pierre McKenzie and Stephen A. Cook. The parallel complexity of the abelian permutation group membership problem. In *24th Annual Symposium on Foundations of Computer Science (sfcs 1983)*, pages 154–161, 1983. doi:10.1109/SFCS.1983.74.

43 Pierre Murdock McKenzie. *Parallel complexity and permutation groups*. PhD thesis, University of Toronto, 1984. AAI0556312.

44 Gary L. Miller. On the $n^{\log n}$ isomorphism technique (a preliminary report). In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, STOC '78, pages 51–58, New York, NY, USA, 1978. Association for Computing Machinery. doi:10.1145/800133.804331.

45 Ketan Mulmuley. A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. *Comb.*, 7(1):101–104, 1987. doi:10.1007/BF02579205.

46 Youming Qiao, Jayalal M. N. Sarma, and Bangsheng Tang. On isomorphism testing of groups with normal Hall subgroups. In *Proc. 28th STACS*, pages 567–578, 2011. doi:10.4230/LIPIcs.STACS.2011.567.

47 David J. Rosenbaum. Bidirectional collision detection and faster deterministic isomorphism testing. arXiv:1304.3935 [cs.DS], 2013.

48 Uwe Schöning. Graph isomorphism is in the low hierarchy. *Journal of Computer and System Sciences*, 37(3):312–323, 1988. doi:10.1016/0022-0000(88)90010-4.

49 Jean-Pierre Serre. *Linear representations of finite groups*, volume Vol. 42 of *Graduate Texts in Mathematics*. Springer-Verlag, New York-Heidelberg, french edition, 1977. doi:10.1007/978-1-4684-9458-7.

50 Michael Sipser. Borel sets and circuit complexity. In David S. Johnson, Ronald Fagin, Michael L. Fredman, David Harel, Richard M. Karp, Nancy A. Lynch, Christos H. Papadimitriou, Ronald L. Rivest, Walter L. Ruzzo, and Joel I. Seiferas, editors, *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 61–69. ACM, 1983. doi:10.1145/800061.808733.

51 Bangsheng Tang. *Towards Understanding Satisfiability, Group Isomorphism and Their Connections*. PhD thesis, Tsinghua University, 2013.

52 D. R. Taunt. Remarks on the isomorphism problem in theories of construction of finite groups. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(1):16–24, 1955. doi:10.1017/S030500410002987X.

53 Jacobo Torán. On the hardness of graph isomorphism. *SIAM J. Comput.*, 33(5):1093–1108, 2004. doi:10.1137/S009753970241096X.

54 Heribert Vollmer. *Introduction to Circuit Complexity - A Uniform Approach*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 1999. doi:10.1007/978-3-662-03927-4.

55 James B. Wilson. The threshold for subgroup profiles to agree is logarithmic. *Theory of Computing*, 15(19):1–25, 2019. doi:10.4086/toc.2019.v015a019.

56 V. N. Zemlyachenko, N. M. Korneenko, and R. I. Tyshkevich. Graph isomorphism problem. *J. Soviet Math.*, 29(4):1426–1481, May 1985. doi:10.1007/BF02104746.