hexagon:2611.01193v2

Exact Matching in Subexponential Time

Dave Goel

First version: November 26, 2026. This version (2.10): November 27, 2026.

Abstract

In 1982 Papadimitriou and Yannakakis posed the *Exact Matching* problem: given a graph whose edges are colored red and blue and an integer k, decide whether the graph contains a perfect matching with exactly k red edges. Mulmuley, Vazirani, and Vazirani placed Exact Matching in randomized polynomial time ($\mathsf{RNC} \subseteq \mathsf{RP}$) in 1987, yet on general n-vertex graphs no unconditional deterministic subexponential ($2^{o(n)}$) algorithm was known. Suppose arithmetic circuits over $\mathbb{Q}$ computing the $\mu \times \mu$ permanent require more than $\mu^{c(\log_2 \mu)^{1/3}}$ gates for some constant $c > 0$ and all sufficiently large μ. While any superpolynomial permanent lower bound already implies deterministic time 2^{s^ε} (for every fixed $\varepsilon > 0$) for low-degree *polynomial identity testing* (PIT)—deciding whether a polynomial given by an arithmetic circuit is identically zero—via Kabanets and Impagliazzo (2004), and 2^{n^ε} for Exact Matching via division-free Pfaffians, working out the quantitative tradeoff yields a strictly faster uniform bound beating 2^{s^ε} for every $\varepsilon > 0$. Writing $\exp_2(t) := 2^t$, we prove under this hypothesis that PIT—given an integer $s \geq 2$ and an integer arithmetic circuit of encoding length at most s computing a polynomial of total degree and coefficient bit-length at most s—runs in deterministic time $\exp_2(\exp_2(K(\log_2 s)^{3/4})) = 2^{s^{o(1)}}$, for a constant $K > 0$ depending only on the lower-bound parameters. On general n-vertex graphs ($n \geq 2$), Exact Matching—together with witness recovery, finding a perfect matching of prescribed total weight for integer edge weights bounded by n^C, and finding a perfect matching of prescribed label sum in any finite abelian group of order at most n^C—then runs in deterministic time $\exp_2(\exp_2(K'(\log_2 n)^{3/4})) = 2^{n^{o(1)}}$ (with K' depending also on C for the weighted and group-labeled variants). Instantiating this hypothesis with the recent permanent lower bound of Claude (Anthropic, November 2026; machine-checked in Lean 4, not yet human-refereed) conditionally places all of these problems in deterministic time $2^{n^{o(1)}}$, and a stronger unrefereed bound of OpenAI conditionally sharpens the Exact Matching time bound to $\exp_2(\exp_2(O((\log_2 n)^{1/2} \log_2 \log_2(n+2))))$. These conditional derandomization bounds combine Kabanets–Impagliazzo hardness–randomness tradeoffs with division-free Pfaffian circuits over integral group rings. Under the stated permanent lower bound, Lean verifies test correctness, the matching reductions and the evaluation count. The running-time bounds are outside this verification.[1]

2020 Mathematics Subject Classification: Primary 68Q15; Secondary 68W20, 05C70, 68Q25.

Keywords: Exact Matching, polynomial identity testing, derandomization, permanent lower bounds, Kabanets–Impagliazzo, Pfaffian, finite abelian groups.

1 Introduction

In 1982 Papadimitriou and Yannakakis [12] introduced the *Exact Matching* problem: given a graph $G = (V, E)$ on n vertices whose edges are colored red and blue (with red edge set $R \subseteq E$) and an integer k, decide whether one can pair up all vertices along disjoint edges—a *perfect matching*—using exactly k red edges, and find such a matching if one exists. They conjectured the problem to be NP-complete. Five years later, Mulmuley, Vazirani, and Vazirani [10] introduced the Isolation Lemma and gave a fast randomized parallel ($\mathsf{RNC} \subseteq \mathsf{RP}$) algorithm for Exact Matching, and Camerini, Galbiati, and Maffioli [2] extended this algebraic method to weighted and matroid-intersection variants. More than four decades later, Exact Matching remains one of the very few natural combinatorial problems solvable quickly with random coin flips (RP) yet not known to admit a fast deterministic algorithm (P). It is highlighted as a central open problem in Svensson and Tarnawski's FOCS 2017 Best Paper [18, Section 1] and by El Maalouly, Steiner, and Wulf [4], much like *Polynomial Identity Testing* (PIT)—deciding whether a compact arithmetic circuit computes the identically zero polynomial (in coRP, randomized polynomial time with one-sided error)—which Shpilka and Yehudayoff [15, Chapter 4], Saxena [14], and Wigderson [19, Chapters 7 and 12] single out as one of the premier open testbeds for derandomization.

[1] Proved in Lean 4 using three standard axioms and checked with `lake comparator`: https://github.com/david44hello/paper_papadimitriou-yannakakis-1982-exact-matching/tree/v2.00/lean

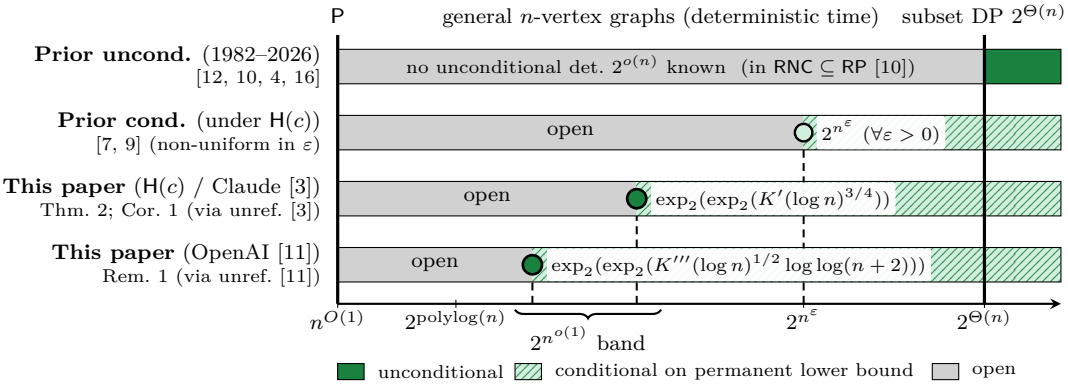

Figure 1: Deterministic time complexity of Exact Matching (and polynomially bounded integer-weight and polynomial-order abelian-group perfect matching) on general n-vertex graphs (schematic monotone scale, $\exp_2(t) = 2^t$). Solid green marks unconditional bounds; hatched green marks bounds conditional on permanent lower bounds; gray marks open regimes. Unconditionally, nothing better than $2^{\Theta(n)}$ subset dynamic programming (DP) was known on general graphs (though $\mathsf{RNC} \subseteq \mathsf{RP}$ has been known since 1987 [10]). Conditionally under $\mathsf{H}(c)$ (the $\mu^{c(\log\mu)^{1/3}}$ permanent lower-bound hypothesis of Section 1), [7] and [9] already give $2^{n^{\varepsilon}}$ for every fixed $\varepsilon > 0$; Theorem 2 uniformly sharpens this into the $2^{n^{o(1)}}$ band at $\exp_2(\exp_2(K'(\log n)^{3/4}))$ (instantiated by unrefereed Claude [3]) and covers arbitrary polynomial-order abelian groups; unrefereed OpenAI [11] sharpens the bound to $\exp_2(\exp_2(K'''(\log n)^{1/2}\log\log(n+2)))$. For weighted ($W \le n^C$) and abelian-group ($|\Gamma| \le n^C$) variants, K', K''' become K'_C, K'''_C (Theorem 2 and Remark 1).

Earlier work

For *ordinary* and *minimum-weight* perfect matching, deterministic algorithms have long been known in P and, more recently, in deterministic quasi-NC (polylogarithmic parallel depth with $2^{(\log n)^{O(1)}}$ processors), first for bipartite graphs [5] and then for general graphs [18]. Those techniques isolate an extremal (minimum-weight) face of the matching polytope—essentially finding the single cheapest perfect matching—and do not access intermediate weight slices such as $|M \cap R| = k$ for a perfect matching M, where the target count lies strictly between the minimum and maximum possible.

For Exact Matching and its modular generalization—finding a perfect matching M whose edge-label sum $w(M)$ in a finite abelian group Γ equals a prescribed target $\gamma^* \in \Gamma$—deterministic polynomial-time algorithms are known only in restricted settings: for $\Gamma = \mathbb{Z}_2$ (prescribed parity of red edges) on bipartite graphs [1] and general graphs [4]; for *avoiding* a single forbidden residue or coset ($w(M) \neq \gamma^*$) on bipartite [6] and general [16] graphs; and on restricted graph families such as planar, bounded-genus, bounded-treewidth, and bounded bipartite-independence-number graphs (see [4] and references therein). On general n-vertex graphs, however, no unconditional deterministic algorithm running in subexponential time $2^{o(n)}$ was known for Exact Matching, nor even for singleton targets in $\Gamma = \mathbb{Z}_3$ or $\Gamma = \mathbb{Z}_2 \times \mathbb{Z}_2$.

On the algebraic side, Kabanets and Impagliazzo [7, Theorem 7.7] proved that any superpolynomial arithmetic circuit lower bound for the permanent already implies deterministic subexponential time $2^{s^{\varepsilon}}$ (for every fixed $\varepsilon > 0$, with thresholds depending on ε) for PIT on low-degree integer circuits, while exponential lower bounds imply quasi-polynomial ($2^{\mathrm{poly}(\log s)}$) PIT. Combined with the division-free Pfaffian circuits (a matching-counting matrix invariant) of Mahajan, Subramanya, and Vinay [9], any superpolynomial permanent lower bound therefore already places Exact Matching and polynomially bounded integer-weight matching in deterministic time $2^{n^{\varepsilon}}$ for every fixed $\varepsilon > 0$. In November 2026, two AI-assisted preprints announced superpolynomial lower bounds for the permanent over every field of characteristic other than 2: Claude (Anthropic) [3, Theorem 1.1] announced a lower bound of $\mu^{c(\log\mu)^{1/3}}$ gates (formalized in Lean 4, not yet human-refereed), explicitly noting in Section 1.2 that they had not worked out what their bound yields for PIT via [7]; and OpenAI [11, Theorem 1.1] announced a stronger bound of $\exp(c(\ln\mu)^2/(\ln\ln\mu)^2)$ gates (likewise not yet human-refereed). Within this conditional-derandomization tier, this note contributes two things: (i) we work out the quantitative tradeoff explicitly (with a finite-field design, an explicit modulus 2^B, and exact circuit-model transfer), uniformly sharpening $2^{s^{\varepsilon}}$ to $2^{s^{o(1)}} = \exp_2(\exp_2(K(\log s)^{3/4}))$ under Hypothesis $\mathsf{H}(c)$ below (instantiated by [3]) and to $\exp_2(\exp_2(K''(\log s)^{1/2}\log\log(s+2)))$ under [11] (for constants $K, K'' > 0$), sitting strictly between the two benchmark regimes of [7, Theorem 7.7]; and (ii) we give a unified division-free Pfaffian reduction over the integral group ring $\mathbb{Z}[\Gamma]$ that avoids factorial-sized interpolation multipliers and handles Exact Matching, bounded integer weights, and arbitrary polynomial-order finite abelian groups Γ in deterministic time $2^{n^{o(1)}}$. These are conditional derandomization bounds in algebraic and combinatorial complexity; they do not bear on P versus NP.

Circuit models and main results

Throughout the paper, all logarithms are to base 2 ($\log := \log_2$), $\exp_2(t) := 2^t$, and $\exp(t) := e^t$. Following [7, Section 2.3], an *arithmetic circuit over* $\mathbb{Z}$ has input variables, integer constants written in binary, and binary gates $+$, $-$, and $\times$; its *encoding length* is the total number of bits required to specify the circuit, which bounds both its number of gates and the total bit-length of all integer constants inside the circuit. (Bounding gate count alone would allow a constant-size circuit to hide arbitrarily large constants that no subexponential algorithm could even read.) Following [3, Section 1.3], an *arithmetic circuit over* $\mathbb{Q}$ is a finite list of formal polynomials in $\mathbb{Q}[x_1, \dots, x_m]$ where each entry is a variable, an arbitrary rational constant $a \in \mathbb{Q}$, or the sum or product of two earlier entries; its *size* is the number of entries (gates), with each constant counted as one gate regardless of bit-length. For $\mu \geq 1$, let $\mathrm{PERM}_\mu(X) = \sum_{\sigma \in S_\mu} \prod_{i=1}^{\mu} X_{i,\sigma(i)}$ be the $\mu \times \mu$ *permanent* polynomial in μ^2 variables—the signless counterpart of the determinant.

Hypothesis $\mathsf{H}(c)$. *For a constant $c > 0$, there exists an integer $\mu_0 \geq 1$ such that for every integer $\mu \geq \mu_0$, every arithmetic circuit over $\mathbb{Q}$ computing PERM_μ has strictly more than $\mu^{c(\log \mu)^{1/3}}$ gates.*

Theorem 1.1 of [3] asserts $\mathsf{H}(c)$ for some absolute constant $c > 0$ over every field of characteristic other than 2, in particular over $\mathbb{Q}$. We state our main results as unconditional implications from $\mathsf{H}(c)$.

Theorem 1 (Quantitative PIT over $\mathbb{Z}$)**.** *Assume $\mathsf{H}(c)$ for some $c > 0$. There is a constant $K > 0$ (depending only on c and μ_0) and a deterministic algorithm that, given an integer $s \geq 2$ and an arithmetic circuit C over $\mathbb{Z}$ of encoding length at most s computing a polynomial f of total degree at most s whose coefficients have bit-length at most s, decides whether $f \equiv 0$ in time $\exp_2(\exp_2(K(\log s)^{3/4}))$.*

Note that $\exp_2(\exp_2(K(\log s)^{3/4})) = 2^{s^{K/(\log s)^{1/4}}} = 2^{s^{o(1)}}$.

Theorem 2 (Exact Matching, bounded integer weights, and abelian group labels)**.** *Assume $\mathsf{H}(c)$ for some $c > 0$. There is a constant $K' > 0$ (depending only on c and μ_0), and for every constant $C > 0$ a constant $K'_C > 0$ (depending only on c, μ_0, and C), such that the following hold on every graph G on $n \geq 2$ vertices:*

(a) Exact Matching. *Given a set $R \subseteq E(G)$ of red edges and an integer $k \in \{0, \dots, \lfloor n/2 \rfloor\}$ (with out-of-range binary targets rejected immediately by a length check), one can decide whether G has a perfect matching with exactly k red edges, and output one if it exists, in deterministic time $\exp_2(\exp_2(K'(\log n)^{3/4})) = 2^{n^{o(1)}}$.*

(b) Bounded integer weights. *Given integer edge weights $w : E(G) \to [-W, W]$ ($W \in \mathbb{Z}_{\geq 0}$) and a target $k \in \mathbb{Z}$ with $|k| \leq \lfloor n/2 \rfloor W$ (out-of-range targets rejected immediately), one can decide whether G has a perfect matching of total weight k, and output one if it exists, in deterministic time $\exp_2(\exp_2(K'(\log(n + W))^{3/4}))$; in particular, in time $\exp_2(\exp_2(K'_C(\log n)^{3/4})) = 2^{n^{o(1)}}$ whenever $W \leq n^C$.*

(c) Finite abelian group labels. *Given a finite abelian group $\Gamma = \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_r}$ (specified by cyclic orders $q_1, \dots, q_r \geq 2$, or $r = 1, q_1 = 1$ for the trivial group $|\Gamma| = 1$), edge labels $w : E(G) \to \Gamma$, and a target $\gamma^* \in \Gamma$, one can decide whether G has a perfect matching with label sum γ^*, and output one if it exists, in deterministic time $\exp_2(\exp_2(K'(\log(n + |\Gamma|))^{3/4}))$; in particular, in time $\exp_2(\exp_2(K'_C(\log n)^{3/4})) = 2^{n^{o(1)}}$ whenever $|\Gamma| \leq n^C$.*

Corollary 1. *Conditional on Theorem 1.1 of [3] over $\mathbb{Q}$, Hypothesis $\mathsf{H}(c)$ holds for some $c > 0$, and hence low-degree integer PIT (Theorem 1), Exact Matching, polynomially bounded integer-weight matching, and polynomial-order finite abelian-group matching on general graphs (Theorem 2) are all solvable in deterministic time $2^{n^{o(1)}}$ (with $n = s$ in Theorem 1).*

Remark 1 (Stronger permanent lower bounds). Suppose Hypothesis $\mathsf{H}(c)$ is replaced by the bound of [11, Theorem 1.1], namely $\exp(c(\ln \mu)^2/(\ln \ln \mu)^2)$ for all $\mu \geq N_0(0)$ (their characteristic-zero size threshold). Note that $(\ln \mu)^2/(\ln \ln \mu)^2 > (\log \mu)^2/(\log \log \mu)^2$ for all $\mu \geq 3$, and every circuit of size $\leq s^\kappa$ from Lemma 2.2 (where $\kappa \geq 1$ is its universal reconstruction exponent) is an exact circuit (an ordinary, non-border arithmetic circuit) of size $\leq s^\kappa$ in [11] (where variables and scalars are free). Fixing a sufficiently large integer $C_1 \geq 4$ (depending on c and κ), taking the least integer $t \geq 1$ with $t^2 \geq C_1^2 \lceil \log s \rceil \lceil \log \log(s + 2) \rceil^2$, and setting $\mu := \max(N_0(0), 2^t)$ then yields deterministic time $\exp_2(\exp_2(K''(\log s)^{1/2} \log \log(s + 2)))$ in Theorem 1 (for all $s \geq 2$) and $\exp_2(\exp_2(K'''(\log N)^{1/2} \log \log(N + 2)))$ in Theorem 2, where $N = n$ in (a), $N = n + W$ in (b), and $N = n + |\Gamma|$ in (c), for constants $K'', K''' > 0$ depending only on the lower-bound parameters. Whenever $W \leq n^C$ in (b) or $|\Gamma| \leq n^C$ in (c), this specializes to $\exp_2(\exp_2(K'''_C(\log n)^{1/2} \log \log(n + 2)))$ for a constant $K'''_C > 0$ depending also on C; the $+2$ shift in $s + 2$ and $N + 2$ prevents the degeneracy $\log \log 2 = 0$ at $s = 2$ or $N = 2$. Because the accompanying Lean 4 files of [11] formalize the qualitative superpolynomial statement rather than the explicit $(\ln \mu)^2/(\ln \ln \mu)^2$ rate, whereas those of [3] formalize

$\mathsf{H}(c)$ in full, we take $\mathsf{H}(c)$ as our primary hypothesis. Neither [3] nor [11] has yet undergone human refereeing; if both were found flawed, Theorems 1 and 2 would remain valid as conditional theorems, while Corollary 1 and the instantiations in this remark would not stand.

Problem (n-vertex graphs / length-s circuits)	Prior uncond.	Prior cond. ($\mathsf{H}(c)$)	This paper (under $\mathsf{H}(c)$)	Randomized
Exact Matching ($\lvert M \cap R\rvert = k$, [12])	$2^{\Theta(n)}$	$2^{n^{\varepsilon}}$ $(\forall \varepsilon > 0)$ [7, 9]	$\exp_2(\exp_2(K'(\log n)^{3/4})) = 2^{n^{o(1)}}$	RNC [10]
Exact-weight ($w(M) = k$, $\lvert w(e)\rvert \le n^C$)	$2^{\Theta(n)}$	$2^{n^{\varepsilon}}$ $(\forall \varepsilon > 0)$ [7, 9]	$\exp_2(\exp_2(K'_C(\log n)^{3/4})) = 2^{n^{o(1)}}$	RNC [10, 2]
Abelian-group ($w(M) = \gamma^*$, $3 \le \lvert\Gamma\rvert \le n^C$)	$2^{\Theta(n)}$	—	$\exp_2(\exp_2(K'_C(\log n)^{3/4})) = 2^{n^{o(1)}}$	RNC/RP (Prop. 3.1)
Low-degree int. PIT (length $\le s$)	$2^{O(s)}$ (simplex)	$2^{s^{\varepsilon}}$ $(\forall \varepsilon > 0)$ [7]	$\exp_2(\exp_2(K(\log s)^{3/4})) = 2^{s^{o(1)}}$	coRP (see [7])

Table 1: Apples-to-apples before-and-after summary for general graphs and low-degree integer circuits, separating unconditional prior bounds (subset DP on graphs; evaluation on the integer simplex $\Delta_s = \{a \in \mathbb{Z}^s_{\ge 0} : \sum_i a_i \le s\}$ of size $\binom{2s}{s} \le 4^s$ for degree-$\le s$ PIT) from prior conditional bounds under $\mathsf{H}(c)$ (which follow non-uniformly in ε from [7, Theorem 7.7(i)] and [9]). For arbitrary polynomial-order abelian groups ($\lvert\Gamma\rvert \le n^C$), unbounded rank (such as $\Gamma = \mathbb{Z}_2^{\lfloor \log n \rfloor}$) blows up naive Kronecker integer-weight encoding to $n^{\Theta(\log n)}$; combining Proposition 3.1 with [7, Theorem 7.7(i)] gives $2^{n^{\varepsilon}}$, while Theorem 2(c) gives $2^{n^{o(1)}}$. For $\Gamma = \mathbb{Z}_2$ parity [4] and avoiding one forbidden residue/coset [16], unconditional deterministic polynomial time was already known. In Row 3, decision lies in RNC via Proposition 3.1 (evaluating balanced-tree convolution sums modulo a small prime $p \in (n, 2n]$) and Schwartz–Zippel [7, Lemma 2.18] (for fixed Γ, randomized polynomial time also follows from [10, 2]), and witness recovery lies in RP by self-reduction; Row 4 lies in coRP by Schwartz–Zippel (see [7, Section 2.4]).

2 Proof of Theorem 1: quantitative PIT from $\mathsf{H}(c)$

Why does hardness of the permanent derandomize polynomial identity testing? Rather than plugging random inputs into a circuit C, we evaluate C on a small deterministic list of pseudorandom tuples generated by applying the permanent to overlapping slices of a short seed; if C computes a nonzero polynomial yet vanishes on every generated tuple, an algebraic factoring step (Kaltofen's algorithm) would turn C into a small circuit for the permanent, contradicting Hypothesis $\mathsf{H}(c)$. Formally, the proof tracks the Nisan–Wigderson generator of Kabanets and Impagliazzo [7, Section 7.2], with an explicit finite-field design, an explicit modulus 2^B preventing intermediate coefficient blowup, and a direct verification of circuit-model compatibility between [7] and [3].

Lemma 2.1 (Explicit polynomial-graph design)**.** *Let $m, n \in \mathbb{Z}_{\ge 1}$ satisfy $\log n < m \le n$ and $n \ge 2$. Let $q = 2^r$ be the least power of 2 with $q \ge m$, and put $l := mq$. Then $l < 2m^2$, and in deterministic $\mathrm{poly}(n)$ time one can construct sets $S_1, \dots, S_n \subseteq [l]$ such that $\lvert S_i\rvert = m$ for all i and $\lvert S_i \cap S_j\rvert < \log n$ for all $i \ne j$.*

Proof. Since $q < 2m$, we have $l = mq < 2m^2$. To construct the finite field $\mathbb{F}_q$ deterministically, test all $2^r = q < 2n$ monic polynomials of degree r over $\mathbb{F}_2$ for divisibility by every non-constant polynomial of degree at most $\lfloor r/2 \rfloor$; this takes $O(q^{3/2}\,\mathrm{poly}(\log q)) = \mathrm{poly}(n)$ time and yields an irreducible polynomial defining $\mathbb{F}_q$. Fix an arbitrary subset $A \subseteq \mathbb{F}_q$ of size m and identify $[l]$ with $A \times \mathbb{F}_q$. Set $t := \lceil \log n / \log q \rceil$. Then $q^t \ge n$, and since $q \ge m > \log n \ge 1$ we have $t - 1 < \log n / \log q \le \log n$. For each univariate polynomial $h \in \mathbb{F}_q[z]$ of degree less than t, define its graph $S_h := \{(u, h(u)) : u \in A\} \subseteq A \times \mathbb{F}_q$. Each S_h has size m. Two distinct polynomials $h \ne h'$ of degree $< t$ agree on at most $t-1$ elements of A, so $\lvert S_h \cap S_{h'}\rvert \le t - 1 < \log n$; because $m > \log n > t - 1$, distinct polynomials produce distinct sets S_h. There are $q^t \ge n$ such polynomials; enumerating the first n in lexicographic order constructs $S_1, \dots, S_n$ in deterministic $\mathrm{poly}(n)$ time. □

For a family $S_1, \dots, S_n \subseteq [l]$ with $\lvert S_i\rvert = m$ and a polynomial $p \in \mathbb{Q}[x_1, \dots, x_m]$, define the Nisan–Wigderson polynomial map $\mathrm{NW}^p : \mathbb{Q}^l \to \mathbb{Q}^n$ by

$$\mathrm{NW}^p(a) := \big(p(a|_{S_1}),\ \dots,\ p(a|_{S_n})\big),$$

where $a|_{S_i} \in \mathbb{Q}^m$ lists the coordinates of a indexed by S_i in increasing order.

Lemma 2.2 (Reconstruction and circuit-model transfer)**.** *There exists an absolute constant $\kappa \ge 1$ with the following property. Let $s \ge 2$, let $f \in \mathbb{Z}[y_1, \dots, y_n]$ be a nonzero polynomial of total degree d_f computable by an arithmetic circuit over $\mathbb{Z}$ of encoding length at most s, and let $p \in \mathbb{Z}[x_1, \dots, x_m]$ be a nonzero multilinear polynomial of total degree d_p with $n \ge 2$ and $\log n < m \le n$. Let b bound the bit-lengths of all coefficients of f and p, and assume $\max(m, n, d_f, d_p, s, b) \le s$. Let $S_1, \dots, S_n \subseteq [l]$ be the sets of Lemma 2.1, and let $S \subseteq \mathbb{Q}$ satisfy $\lvert S\rvert > d_f d_p$. If $f(\mathrm{NW}^p(a)) = 0$ for every $a \in S^l$, then p is computable by an arithmetic circuit over $\mathbb{Q}$ (in the sense of [3, Section 1.3]) with at most s^{κ} gates.*

Proof. If $d_p = 0$, then p is a nonzero integer constant, computable by a single gate ($1 \le s^\kappa$). Assume $d_p \ge 1$. This lemma is then [7, Lemma 7.6, pp. 35–36] over $\mathbb{F} = \mathbb{Q}$, combined with the model translation into [3, Section 1.3]; we check the exact bit-length and model steps carefully.

Since the formal polynomial $f(\mathrm{NW}^p(x_1, \ldots, x_l)) \in \mathbb{Z}[x_1, \ldots, x_l]$ has total degree at most $d_f d_p < |S|$ and vanishes at every $a \in S^l$, multivariate polynomial nonvanishing [7, Lemma 2.18] implies the formal identity $f(\mathrm{NW}^p(x)) \equiv 0$. Define the hybrid polynomials $g_0(y_1, \ldots, y_n) := f(y_1, \ldots, y_n)$ and, for $1 \le j \le n$,

$$g_j(x_1, \ldots, x_l, y_{j+1}, \ldots, y_n) := f\big(p(x|_{S_1}), \ldots, p(x|_{S_j}), y_{j+1}, \ldots, y_n\big).$$

Because $g_0 \not\equiv 0$ and $g_n \equiv 0$, there is a least index $i \in \{0, \ldots, n-1\}$ with $g_i \not\equiv 0$ and $g_{i+1} \equiv 0$ (forcing $d_f \ge 1$). Since $\deg(g_i) \le d_f d_p$, the integer grid $S_0 := \{0, 1, \ldots, d_f d_p\}$ has size $d_f d_p + 1 > \deg(g_i)$; hence by [7, Lemma 2.18] we can fix all variables x_k ($k \notin S_{i+1}$) and $y_{i+2}, \ldots, y_n$ to integers in S_0 so that the resulting polynomial $g(x_1, \ldots, x_m, y)$ (reindexing $x|_{S_{i+1}}$ as $x_1, \ldots, x_m$ and y_{i+1} as y) is nonzero of degree at most $d_f d_p$, while $g(x_1, \ldots, x_m, p(x_1, \ldots, x_m)) \equiv 0$.

For each $j \le i$, Lemma 2.1 gives $|S_j \cap S_{i+1}| < \log n$, so the restricted multilinear polynomial $p_j(x_1, \ldots, x_m)$ obtained by substituting values from S_0 outside S_{i+1} has at most $2^{\log n} = n$ monomials. Because p is multilinear in m variables (at most 2^m monomials) of degree d_p with coefficients of magnitude at most 2^b, each coefficient of p_j has magnitude at most $M_p := 2^{m+b}(d_f d_p)^{d_p}$, of bit-length $O(m + b + d_p \log(d_f d_p + 1)) = \mathrm{poly}(m, n, d_f, d_p, s, b)$. Substituting these $i \le n$ explicit polynomials and the constants from S_0 into the circuit for f yields a circuit over $\mathbb{Z}$ for g of size (including constant bit-lengths) at most $\mathrm{poly}(m, n, d_f, d_p, s, b)$. Moreover, since f has at most $\binom{n+d_f}{d_f} \le (n+1)^{d_f}$ monomials of degree at most d_f with coefficients of magnitude at most 2^b and each p_j has at most n monomials with coefficients of magnitude at most M_p, every coefficient of g has magnitude at most $(n+1)^{d_f} \cdot 2^b \cdot (nM_p)^{d_f}$, whose bit-length is $O(b + d_f(\log n + m + b + d_p \log(d_f d_p + 1))) = \mathrm{poly}(m, n, d_f, d_p, s, b)$. Applying Kaltofen's factoring theorem [8] via [7, Corollary 7.4, p. 34] produces a circuit over $\mathbb{Q}$ computing p of size $\mathrm{poly}(m, n, d_f, d_p, s, b)$ (with a universal polynomial independent of the instance; if a nonzero scalar multiple $c(y - p(x))$ is returned, setting $y = 0$ and multiplying by $-c^{-1}$ adds at most 2 gates).

To pass into the circuit model of [3, Section 1.3], observe three points. First, Corollary 7.4 of [7] outputs a division-free circuit over $\mathbb{Q}$ (and even if divisions occurred, Strassen's division-elimination theorem [17]—see [7, Theorem 2.17]—replaces them at polynomial overhead by divisions by nonzero constants only). Second, in [3, Section 1.3] every rational constant $a \in \mathbb{Q}$ is a single gate regardless of bit-length, so each subtraction $u - v = u + (-1) \cdot v$ uses at most 3 gates and each scalar multiplication $c \cdot u$ or constant division $c^{-1} \cdot u$ ($c \in \mathbb{Q}^\times$) uses at most 2 gates, increasing gate count by at most a factor of 3. Third, because $\mathbb{Q}$ is infinite, a circuit computing p functionally on $\mathbb{Q}^m$ computes p as a formal polynomial in $\mathbb{Q}[x_1, \ldots, x_m]$ [7, pp. 12–13]. Absorbing the constant factor into the universal polynomial bound gives at most s^κ gates whenever $\max(m, n, d_f, d_p, s, b) \le s$ and $s \ge 2$. □

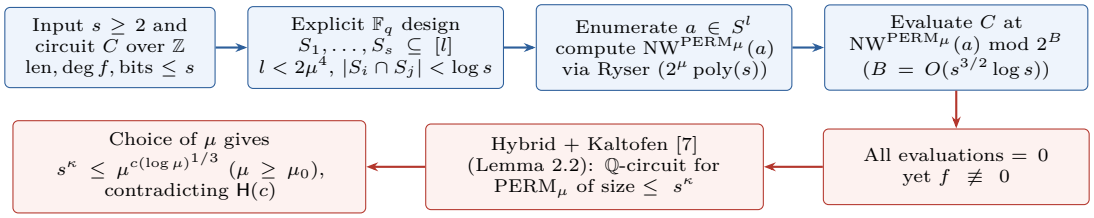

Figure 2: Architecture of Theorem 1, with $\mu = \max(\mu_0, 2^t) = 2^{O((\log s)^{3/4})}$ and grid $S = \{0, \ldots, s\mu\}$. Top row: the deterministic algorithm runs in time $|S|^l \cdot 2^\mu \cdot \mathrm{poly}(s) \le \exp_2(\exp_2(K(\log s)^{3/4}))$. Bottom row: if $f \not\equiv 0$ escapes the test, Lemma 2.2 contradicts Hypothesis $\mathsf{H}(c)$.

Proof of Theorem 1. Fix the constants $c > 0$, $\mu_0 \in \mathbb{Z}_{\ge 1}$ from $\mathsf{H}(c)$ and $\kappa \ge 1$ from Lemma 2.2, and fix the single integer constant $A := \lceil \kappa / c \rceil \in \mathbb{Z}_{\ge 1}$. Given an integer $s \ge 2$ and an arithmetic circuit C over $\mathbb{Z}$ of encoding length at most s computing f of total degree $d_f \le s$ with coefficients of bit-length at most s, each variable referenced by C contributes to its encoding length, so f depends on at most s variables; adding dummy variables, we view f as a polynomial in $n = s$ variables.

Let $L := \lceil \log s \rceil \in \mathbb{Z}_{\ge 1}$ (the bit-length of $s - 1$), let $t \in \mathbb{Z}_{\ge 1}$ be the least integer satisfying $t^4 \ge (AL)^3$ (found by exact integer arithmetic in $\mathrm{poly}(\log s)$ time), and define

$$\mu := \max(\mu_0,\ 2^t).$$

Set $m := \mu^2$ and $p := \mathrm{PERM}_\mu$, a multilinear polynomial in m variables of degree $d_p = \mu$ with coefficients in $\{0,1\}$ (bit-length $1 \le s$). By choice of μ, we have $\mu \ge \mu_0$ and $(\log \mu)^{4/3} \ge t^{4/3} \ge AL \ge (\kappa/c)\log s$, whence

$$\mu^{c(\log \mu)^{1/3}} = \exp_2\big(c(\log \mu)^{4/3}\big) \ge \exp_2(\kappa \log s) = s^\kappa. \tag{1}$$

Moreover, $(t-1)^4 < (AL)^3$ implies $t < (A(\log s + 1))^{3/4} + 1 \le (A \log s)^{3/4} + A^{3/4} + 1$ (since $(u+v)^{3/4} \le u^{3/4} + v^{3/4}$), so $\mu = O(\exp_2((A \log s)^{3/4}))$ and $m = \mu^2 = 2^{O((\log s)^{3/4})} = s^{o(1)}$. Hence there exists an integer threshold $s_0 = s_0(A, \mu_0) \ge 4$ such that for all $s \ge s_0$ we have $\log s < m < s$ and $s\mu + 1 \le s^2$. (The algorithm is completely uniform, depending only on the three fixed integer constants A, μ_0, s_0.)

Small inputs ($s < s_0$). Evaluate C directly on every grid point $a \in \{0, 1, \dots, s\}^s$ modulo 2^{B_0}, where $B_0 := 3s + s\lceil \log s \rceil + 2$. Since f has at most $\binom{2s}{s} \le 2^{2s}$ monomials of degree at most s and coefficients of magnitude at most 2^s, at every $a \in \{0, \dots, s\}^s$ we have $|f(a)| \le 2^{2s} \cdot 2^s \cdot s^s = 2^{3s + s\log s} < 2^{B_0 - 1}$. Thus $f(a) \equiv 0 \pmod{2^{B_0}}$ if and only if $f(a) = 0$, and by standard multivariate polynomial nonvanishing on a grid of side $s + 1 > d_f$, $f \equiv 0$ if and only if all $(s+1)^s$ values vanish. Because $s < s_0$, this takes deterministic $O_{c,\kappa,\mu_0}(1)$ time.

Large inputs ($s \ge s_0$). Construct the set system $S_1, \dots, S_s \subseteq [l]$ of Lemma 2.1 with $n = s$, $m = \mu^2$, and $l < 2m^2 = 2\mu^4$. Set $S := \{0, 1, \dots, s\mu\}$, so $|S| = s\mu + 1 > d_f d_p$ and $|S| \le s^2$, and set

$$B := 3s + s\mu\lceil \log(s\mu^2) \rceil + 2 = O(s^{3/2} \log s).$$

The algorithm enumerates every $a \in S^l$. For each a:

1. For each $i = 1, \dots, s$, compute $v_i := \mathrm{PERM}_\mu(a|_{S_i})$ (viewing $a|_{S_i}$ as a $\mu \times \mu$ matrix with entries in $\{0, \dots, s\mu\}$) using Ryser's formula [13]:
$$\mathrm{PERM}_\mu(M) = (-1)^\mu \sum_{T \subseteq [\mu]} (-1)^{|T|} \prod_{r=1}^{\mu} \sum_{j \in T} M_{r,j},$$
which performs $O(2^\mu \mu^2)$ arithmetic operations on integers of $O(\mu \log(s\mu))$ bits, hence $2^\mu \,\mathrm{poly}(\mu, \log s)$ bit operations.
2. Evaluate the circuit C at $v = (v_1, \dots, v_s)$ modulo 2^B (reducing each input v_i and each binary constant of C modulo 2^B, and carrying out all $+, -, \times$ gates in $\mathbb{Z}/2^B\mathbb{Z}$).

Output "$f \equiv 0$" if every $a \in S^l$ gives $0 \pmod{2^B}$, and "$f \not\equiv 0$" otherwise.

Correctness. Each coordinate of $v = \mathrm{NW}^{\mathrm{PERM}_\mu}(a)$ satisfies $0 \le v_i \le \mu!(s\mu)^\mu \le (s\mu^2)^\mu$. Since f has s variables, degree at most s, at most $\binom{2s}{s} \le 2^{2s}$ monomials, and coefficients of magnitude at most 2^s, we obtain

$$|f(v)| \le 2^{2s} \cdot 2^s \cdot \big((s\mu^2)^\mu\big)^s = 2^{3s + s\mu \log(s\mu^2)} < 2^{B-1}.$$

Because reduction modulo 2^B is a ring homomorphism $\mathbb{Z} \to \mathbb{Z}/2^B\mathbb{Z}$, the evaluation at v equals $f(v) \bmod 2^B$, which is 0 if and only if $f(v) = 0$ (even though intermediate gates of C may compute integers of doubly exponential magnitude, as in [7, Theorem 7.7]). If $f \equiv 0$, every evaluation is 0. Conversely, if $f \not\equiv 0$ and every evaluation is 0, then $f(\mathrm{NW}^{\mathrm{PERM}_\mu}(a)) = 0$ for all $a \in S^l$. Taking $b := s$, we have $\max(m, n, d_f, d_p, s, b) \le s$, so Lemma 2.2 gives an arithmetic circuit over $\mathbb{Q}$ computing PERM_μ with at most s^κ gates. By (1), this circuit has at most $\mu^{c(\log \mu)^{1/3}}$ gates with $\mu \ge \mu_0$, contradicting Hypothesis $\mathsf{H}(c)$.

Running time. Constructing $S_1, \dots, S_s$ takes $\mathrm{poly}(s)$ time. For each of the $|S|^l \le (s^2)^{2\mu^4} = 2^{4\mu^4 \log s}$ tuples $a \in S^l$, computing $\mathrm{NW}^{\mathrm{PERM}_\mu}(a)$ and evaluating C modulo 2^B takes $s \cdot 2^\mu \,\mathrm{poly}(s) + \mathrm{poly}(s, B) = 2^\mu \,\mathrm{poly}(s)$ time. Hence the total deterministic time for $s \ge s_0$ is bounded by $2^{4\mu^4 \log s + \mu + O(\log s)}$. Since $\mu = O(\exp_2((A \log s)^{3/4}))$, we have

$$4\mu^4 \log s + \mu + O(\log s) = O\Big(\exp_2\big(4A^{3/4}(\log s)^{3/4} + \log\log s\big)\Big) \le \exp_2\big(K(\log s)^{3/4}\big)$$

for a constant $K > 0$ depending only on c, κ, μ_0 (chosen large enough to cover the $O_{c,\kappa,\mu_0}(1)$ time for $s < s_0$ as well). Thus the algorithm runs in deterministic time $\exp_2(\exp_2(K(\log s)^{3/4}))$. □

3 Proof of Theorem 2: Exact Matching and abelian group labels

How does finding a perfect matching with a prescribed red-edge count or label sum reduce to testing whether a polynomial is identically zero? Assigning each edge e a formal variable x_e tagged by its group label $w(e)$ inside a skew-symmetric matrix turns the Pfaffian into a generating polynomial whose γ^*-component sums distinct squarefree monomials, one for each valid matching of label sum γ^*; because distinct matchings yield distinct monomials, no cancellation can occur, so that polynomial is nonzero if and only if a valid matching exists. We unify all three parts of Theorem 2 through a single division-free Pfaffian circuit over the integral group ring $\mathbb{Z}[\Gamma]$.

Proposition 3.1 (Group-ring Pfaffian circuit)**.** *Let $G = (\{1,\dots,n\}, E)$ be a graph with $n \geq 2$ even, let $\Gamma = \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_r}$ be a finite abelian group (given by cyclic orders $q_1,\dots,q_r \geq 2$, or $r = 1, q_1 = 1$ when $|\Gamma| = 1$), let $w : E \to \Gamma$ be an edge labeling, and let $\gamma^* \in \Gamma$. In deterministic* $\mathrm{poly}(n, |\Gamma|)$ *time one can construct an arithmetic circuit C_{γ^*} over $\mathbb{Z}$, using only constants in $\{0, \pm 1\}$, of encoding length $s = \mathrm{poly}(n, |\Gamma|)$, that computes a polynomial $P_{\gamma^*}(x) \in \mathbb{Z}[\{x_e : e \in E\}]$ of total degree at most $n/2$ (homogeneous of degree $n/2$ when nonzero) with all coefficients in $\{-1, 0, 1\}$ such that $P_{\gamma^*} \not\equiv 0$ if and only if G has a perfect matching M with $\sum_{e\in M} w(e) = \gamma^*$.*

Proof. Let $A := \mathbb{Z}[\Gamma]$ be the integral group ring of Γ, a commutative ring with $\mathbb{Z}$-basis $\{e_\gamma : \gamma \in \Gamma\}$, multiplicative identity e_0, and multiplication $e_\alpha e_\beta = e_{\alpha+\beta}$. Introduce one indeterminate x_e for each edge $e \in E$, and form the $n \times n$ skew-symmetric *Tutte matrix* T over $A[x]$ defined for $u < v$ by

$$T_{uv} := \begin{cases} x_e\, e_{w(e)} & \text{if } e = \{u, v\} \in E, \\ 0 & \text{otherwise,} \end{cases} \qquad T_{vu} := -T_{uv}, \qquad T_{uu} := 0.$$

The Pfaffian of a skew-symmetric $n \times n$ matrix of indeterminates is the universal integer polynomial $\mathrm{Pf}(Z) = \sum_M \mathrm{sgn}(M) \prod_{\{u,v\}\in M,\, u<v} Z_{uv}$, where the sum runs over all perfect matchings M of the complete graph K_n and $\mathrm{sgn}(M) \in \{\pm 1\}$ is the sign of the matching permutation. Evaluating this universal identity in the commutative ring $A[x]$ gives

$$\mathrm{Pf}(T) = \sum_{M\in\mathcal{M}(G)} \mathrm{sgn}(M) \left(\prod_{e\in M} x_e\right) e_{w(M)} = \sum_{\gamma\in\Gamma} P_\gamma(x)\, e_\gamma,$$

where $\mathcal{M}(G)$ is the set of perfect matchings of G, $w(M) := \sum_{e\in M} w(e) \in \Gamma$, and

$$P_\gamma(x) := \sum_{\substack{M\in\mathcal{M}(G) \\ w(M)=\gamma}} \mathrm{sgn}(M) \prod_{e\in M} x_e \in \mathbb{Z}[x].$$

Because distinct perfect matchings $M \neq M'$ have distinct edge sets, their squarefree monomials $\prod_{e\in M} x_e$ are pairwise distinct and cannot cancel. Hence every coefficient of $P_{\gamma^*}(x)$ lies in $\{-1, 0, 1\}$, $P_{\gamma^*}(x)$ has total degree at most $n/2$ (homogeneous of degree $n/2$ when nonzero), and $P_{\gamma^*} \not\equiv 0$ if and only if there exists $M \in \mathcal{M}(G)$ with $w(M) = \gamma^*$.

By Mahajan, Subramanya, and Vinay [9], the Pfaffian of an $n \times n$ skew-symmetric matrix has a uniform (NC, hence deterministically $\mathrm{poly}(n)$-time constructible) division-free arithmetic circuit of size $\mathrm{poly}(n)$ using only $+$, $-$, $\times$, and constants $0, \pm 1$. Because it is division-free over $\mathbb{Z}$, it computes the universal formal polynomial $\mathrm{Pf}(Z) \in \mathbb{Z}[Z]$ and therefore evaluates identically to $\mathrm{Pf}(T)$ under the ring homomorphism $\mathbb{Z}[Z] \to A[x]$ sending $Z_{uv} \mapsto T_{uv}$, even though $A = \mathbb{Z}[\Gamma]$ has zero divisors when $|\Gamma| \geq 2$.

We expand this $A[x]$-circuit into a circuit over $\mathbb{Z}[x]$ by representing each element $u = \sum_{\gamma\in\Gamma} u_\gamma(x)\, e_\gamma \in A[x]$ as a tuple of $|\Gamma|$ wires $(u_\gamma)_{\gamma\in\Gamma}$:

- An input entry $T_{uv} = x_e e_{w(e)}$ places x_e on wire $w(e)$ and 0 on the other $|\Gamma| - 1$ wires (and $-x_e$ on wire $w(e)$ for T_{vu}).
- A scalar constant $c \in \{0, \pm 1\}$ places c on the identity wire 0 (corresponding to e_0) and 0 on the remaining $|\Gamma| - 1$ wires.
- An addition $u + v$ (respectively subtraction $u - v$) in $A[x]$ is implemented coordinatewise by $|\Gamma|$ additions (respectively subtractions) in $\mathbb{Z}[x]$.
- A multiplication $u \cdot v$ in $A[x]$ is convolution over Γ, $(u \cdot v)_\gamma = \sum_{\alpha+\beta=\gamma} u_\alpha v_\beta$, which uses $|\Gamma|^2$ multiplications and $|\Gamma|(|\Gamma| - 1)$ additions in $\mathbb{Z}[x]$ (arranged as a balanced binary addition tree of depth $\lceil \log |\Gamma| \rceil$, preserving NC depth when $|\Gamma| \leq n^C$).

Because $q_1,\dots,q_r \geq 2$ (or $r = 1, q_1 = 1$), we have $r \leq \max(1, \log|\Gamma|)$, so every group element has bit-length $O(\log(|\Gamma| + 1))$ and enumerating all $|\Gamma| = \prod_{i=1}^r q_i$ elements and their group sums $\alpha + \beta$ takes $\mathrm{poly}(|\Gamma|)$ time. Selecting the output wire indexed by γ^* yields the desired circuit C_{γ^*} over $\mathbb{Z}$ with constants in $\{0, \pm 1\}$, gate count $O(|\Gamma|^2 \,\mathrm{poly}(n))$, and encoding length $s = O(|\Gamma|^2 \log(|\Gamma| + 1)\,\mathrm{poly}(n)) = \mathrm{poly}(n, |\Gamma|)$. □

Proof of Theorem 2. If n is odd, G has no perfect matching and we immediately output "no." Assume henceforth that $n \geq 2$ is even.

Part (c) *(Finite abelian group labels).* Construct the circuit C_{γ^*} of Proposition 3.1 in deterministic $\mathrm{poly}(n, |\Gamma|)$ time, and let $s \geq \max(n, 2)$ bound its encoding length ($s = \mathrm{poly}(n, |\Gamma|)$), which also bounds its degree $n/2$ and

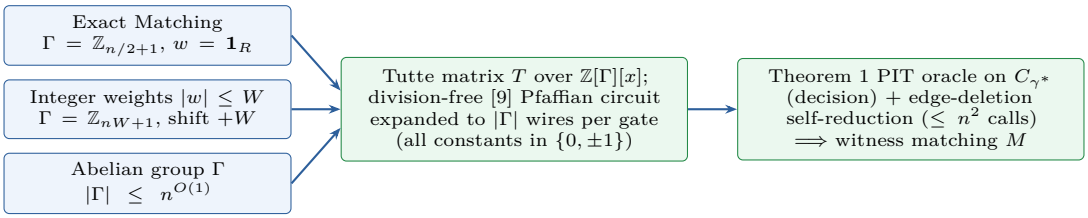

Figure 3: Unified proof of Theorem 2. Exact Matching (with indicator label $w = \mathbf{1}_R$) and polynomially bounded integer weights embed directly into cyclic groups $\mathbb{Z}_{n/2+1}$ and $\mathbb{Z}_{nW+1}$, allowing Proposition 3.1 to produce a $\{0, \pm 1\}$-constant circuit for $P_{\gamma^*}(x)$ in all three cases.

coefficient bit-length 1. Applying Theorem 1 to (s, C_{γ^*}) decides whether $P_{\gamma^*} \not\equiv 0$—and hence whether G has a perfect matching of label sum γ^*—in total deterministic time

$$\mathrm{poly}(n, |\Gamma|) + \exp_2\big(\exp_2(K(\log s)^{3/4})\big) \le \exp_2\big(\exp_2(K_1(\log(n + |\Gamma|))^{3/4})\big),$$

for a constant $K_1 > 0$ absorbing both $\log s = O(\log(n + |\Gamma|))$ and the $\mathrm{poly}(n, |\Gamma|)$ circuit-construction time.

To recover a witness matching when the decision answer is "yes," initialize $H := G$ and iterate once over all edges $e \in E(G)$: test whether $H \setminus \{e\}$ still has a perfect matching of label sum γ^*; if so, update $H \leftarrow H \setminus \{e\}$, and otherwise retain e. By induction, H always contains at least one valid perfect matching M. When the loop terminates, we claim $E(H) = M$. Indeed, if some edge $e \in E(H) \setminus M$ remained, then at the step when e was tested, the current graph $H' \supseteq H$ contained M, which avoids e; thus $H' \setminus \{e\}$ admitted the valid matching M and e would have been deleted, a contradiction. Hence the final edge set $E(H)$ is a valid perfect matching with label sum γ^*. This self-reduction makes at most $|E| + 1 \le n^2$ calls to the decision procedure, and multiplying by n^2 is absorbed into a slightly larger constant $K_2 > K_1$, giving time at most $\exp_2(\exp_2(K_2(\log(n + |\Gamma|))^{3/4}))$.

Part (a) *(Exact Matching).* If $k < 0$ or $k > n/2$, output "no." Otherwise apply part (c) with the cyclic group $\Gamma := \mathbb{Z}_{n/2+1}$ ($|\Gamma| = n/2 + 1 \ge 2$), labels $w(e) := 1$ for $e \in R$ and $w(e) := 0$ for $e \notin R$, and target $\gamma^* := k \bmod (n/2+1)$. Since every perfect matching M has $n/2$ edges, its red-edge count $|M \cap R|$ lies in $\{0, 1, \ldots, n/2\}$, so $|M \cap R| = k$ if and only if $w(M) = \gamma^*$ in $\mathbb{Z}_{n/2+1}$. Moreover, $n + (n/2 + 1) \le n^2$ for all $n \ge 2$, so $\log(n + |\Gamma|) \le 2 \log n$.

Part (b) *(Bounded integer weights).* Reject immediately by a binary length check if $|k| > (n/2)W$. Otherwise shift every edge weight to $w'(e) := w(e) + W \in \{0, 1, \ldots, 2W\}$ and shift the target to $k' := k + (n/2)W \in \{0, 1, \ldots, nW\}$, and apply part (c) with $\Gamma := \mathbb{Z}_{nW+1}$ ($|\Gamma| = nW + 1$) and target $\gamma^* := k' \bmod (nW + 1)$. Every perfect matching M satisfies $0 \le w'(M) \le (n/2)(2W) = nW$, so $w(M) = k$ if and only if $w'(M) = k'$, which holds if and only if $w'(M) \equiv \gamma^* \pmod{nW + 1}$. Since $n + (nW + 1) \le (n + W)^2$ for all $n \ge 2$ and $W \ge 0$, we have $\log(n + |\Gamma|) \le 2 \log(n + W)$.

Choosing a constant $K' > 2^{3/4} K_2$ sufficiently large (absorbing also the $\mathrm{poly}(n)$ and $\mathrm{poly}(n, \log(W+2))$ preprocessing overhead of parts (a) and (b)) yields the stated uniform bound for all three parts (a)–(c); whenever $W \le n^C$ or $|\Gamma| \le n^C$, $\log(n + W)$ and $\log(n + |\Gamma|)$ are $O_C(\log n)$, yielding $\exp_2(\exp_2(K'_C(\log n)^{3/4})) = 2^{n^{o(1)}}$. □

Remark 2 (Alternative interpolation proof for parts (a) and (b)). For Exact Matching and bounded integer weights, one can alternatively define $T_{uv} = x_e y^{w'(e)}$ over $\mathbb{Z}[x, y]$ (forming $j^{w'(e)}$ by repeated squaring) and recover $D \cdot P_{k'}(x)$ with $D = (nW)!$ by Lagrange interpolation of $\mathrm{Pf}(T)|_{y=j}$ at $j = 0, 1, \ldots, nW$, yielding circuits with $O(nW \log(nW + 2))$-bit integer multipliers. The group-ring reduction above avoids factorials altogether, keeps all circuit constants in $\{0, \pm 1\}$, and handles arbitrary polynomial-order finite abelian groups in the same framework.

Discussion

Before this note, no unconditional deterministic $2^{o(n)}$-time algorithm was known on general n-vertex graphs for Papadimitriou and Yannakakis's Exact Matching problem [12] or for prescribed-sum perfect matching in finite abelian groups of order $|\Gamma| \ge 3$. Conditionally under any superpolynomial permanent lower bound, Kabanets and Impagliazzo [7, Theorem 7.7(i)] already gave deterministic time 2^{s^ε} (for every fixed $\varepsilon > 0$) for low-degree integer PIT, and combining this with division-free Pfaffians [9] already gave 2^{n^ε} for Exact Matching and bounded integer weights, while Anthropic's November 2026 permanent lower bound [3] left the quantitative PIT tradeoff unworked. Theorems 1 and 2 supply the explicit uniform $2^{s^{o(1)}}$ tradeoff together with a unified group-ring Pfaffian reduction covering arbitrary polynomial-order finite abelian groups, conditionally placing low-degree integer PIT and Exact Matching (with witnesses, bounded integer weights, and abelian group labels) in deterministic time $2^{n^{o(1)}}$ under Hypothesis $\mathsf{H}(c)$. Corollary 1 and Remark 1 rest on the unrefereed November 2026 permanent lower bounds [3, 11].

How far deterministic time can be pushed beyond $2^{n^{o(1)}}$ depends both on stronger permanent lower bounds and on new derandomization ideas. Under $\mathsf{H}(c)$ (instantiated by [3]) our running time is $\exp_2(\exp_2(K'(\log n)^{3/4}))$, and under [11] it improves to $\exp_2(\exp_2(K'''(\log n)^{1/2}\log\log(n+2)))$. Proving an exponential permanent lower bound $2^{\mu^{\Omega(1)}}$ would place low-degree integer PIT and Exact Matching in deterministic quasi-polynomial time $2^{\mathrm{poly}(\log n)}$ via [7, Theorem 7.7(ii)], whereas placing Exact Matching in deterministic P (even for $\Gamma=\mathbb{Z}_3$ or on bipartite graphs) remains open and appears to lie beyond black-box Nisan–Wigderson derandomization [7, Section 7.3, p. 38]. Meanwhile, independent human refereeing of [3] and [11] is needed to make Corollary 1 and Remark 1 unconditional, and formalizing the full quantitative $\exp(c(\ln\mu)^2/(\ln\ln\mu)^2)$ rate of [11] in Lean 4 would put that sharper bound on the same machine-checked footing as $\mathsf{H}(c)$.

Finally, Proposition 3.1 relies on commutativity of $\mathbb{Z}[\Gamma]$ so that the division-free Pfaffian circuit evaluates validly over $\mathbb{Z}[\Gamma][x]$, and requires $|\Gamma|$ (or W) to be polynomially bounded so that the expanded circuit has polynomial encoding length. Extending deterministic subexponential algorithms to non-abelian group labels, or to binary-encoded edge weights $W=2^{\Omega(n)}$ (where even randomized polynomial time remains unknown for Exact Matching), is left open.

Acknowledgments

This work was done overwhelmingly by AI models, steered by the author in a continuous critical dialogue, variously with Anthropic's Claude, OpenAI's ChatGPT and Google's Gemini. The author takes full responsibility for the paper.

References

[1] S. Artmann, R. Weismantel, and R. Zenklusen, *A strongly polynomial algorithm for bimodular integer linear programming*, Proc. 49th ACM Symposium on Theory of Computing (STOC), 2017, 1206–1219, https://doi.org/10.1145/3055399.3055473

[2] P. M. Camerini, G. Galbiati, and F. Maffioli, *Random pseudo-polynomial algorithms for exact matroid problems*, J. Algorithms **13** (1992), 258–273, https://doi.org/10.1016/0196-6774(92)90018-8

[3] Claude (Anthropic), VP ≠ VNP *over every field*, preprint and Lean 4 formalization, November 23, 2026, https://github.com/anthropics/formal-math/tree/main/vp-vnp

[4] N. El Maalouly, R. Steiner, and L. Wulf, *Exact matching: correct parity and FPT parameterized by independence number*, 34th International Symposium on Algorithms and Computation (ISAAC 2023), LIPIcs **283**, 28:1–28:18, https://doi.org/10.4230/LIPIcs.ISAAC.2023.28

[5] S. Fenner, R. Gurjar, and T. Thierauf, *Bipartite perfect matching is in quasi-NC*, Proc. 48th ACM Symposium on Theory of Computing (STOC), 2016, 754–763; journal version in SIAM J. Comput. **50** (2021), STOC16-218–STOC16-235, https://doi.org/10.1137/16M1097870

[6] F. Hörsch, A. Imolay, R. Mizutani, T. Oki, and T. Schwarcz, *Problems on group-labeled matroid bases*, 51st International Colloquium on Automata, Languages, and Programming (ICALP 2024), LIPIcs **297**, 86:1–86:20, https://doi.org/10.4230/LIPIcs.ICALP.2024.86

[7] V. Kabanets and R. Impagliazzo, *Derandomizing polynomial identity tests means proving circuit lower bounds*, Comput. Complexity **13** (2004), 1–46, https://doi.org/10.1007/s00037-004-0182-6

[8] E. Kaltofen, *Factorization of polynomials given by straight-line programs*, in *Randomness and Computation* (S. Micali, ed.), Adv. Comput. Res. **5**, JAI Press, 1989, 375–412.

[9] M. Mahajan, P. R. Subramanya, and V. Vinay, *The combinatorial approach yields an NC algorithm for computing Pfaffians*, Discrete Appl. Math. **143** (2004), 1–16, https://doi.org/10.1016/j.dam.2003.12.001

[10] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani, *Matching is as easy as matrix inversion*, Combinatorica **7** (1987), 105–113, https://doi.org/10.1007/BF02579206

[11] OpenAI, *Circuit lower bounds for Hamiltonian cycle and the permanent*, preprint and Lean 4 formalization, November 24, 2026, https://github.com/openai/math/tree/main/preprints/Circuit-lower-bounds-for-Hamiltonian-cycle-and-the-permanent-November-24-2026

[12] C. H. Papadimitriou and M. Yannakakis, *The complexity of restricted spanning tree problems*, J. Assoc. Comput. Mach. **29** (1982), 285–309, https://doi.org/10.1145/322307.322309

[13] H. J. Ryser, *Combinatorial Mathematics*, Carus Mathematical Monographs **14**, Mathematical Association of America, 1963.

[14] N. Saxena, *Progress on polynomial identity testing—II*, in *Perspectives in Computational Complexity* (M. Agrawal and V. Arvind, eds.), Progr. Comput. Sci. Appl. Logic **26**, Birkhäuser/Springer, 2014, 131–146, https://doi.org/10.1007/978-3-319-05446-9_7

[15] A. Shpilka and A. Yehudayoff, *Arithmetic circuits: a survey of recent results and open questions*, Found. Trends Theor. Comput. Sci. **5** (2010), 207–388, https://doi.org/10.1561/0400000039

[16] O. Silina, *A polyhedral perspective on the perfect matching lattice*, preprint, November 2025, https://arxiv.org/abs/2511.03863

[17] V. Strassen, *Vermeidung von Divisionen*, J. Reine Angew. Math. **264** (1973), 184–202, https://doi.org/10.1515/crll.1973.264.184

[18] O. Svensson and J. Tarnawski, *The matching problem in general graphs is in quasi-NC*, Proc. 58th IEEE Symposium on Foundations of Computer Science (FOCS), 2017, 696–707, https://doi.org/10.1109/FOCS.2017.70

[19] A. Wigderson, *Mathematics and Computation: A Theory Revolutionizing Technology and Science*, Princeton University Press, 2019.