FerQ: Fermat Quotient Reformulation of High-Order Binary Optimisation

Phuong-Nam Nguyen[1], **Seng W. Loke**[1], **Anil Prabhakar**[2], **Son N. Tran**[1]
[1]Faculty of Science, Engineering and Built Environment, School of Information Technology, Deakin University, 221 Burwood Highway, Burwood VIC 3125, Australia
[2]Department of Electrical Engineering, Indian Institute of Technology Madras, IIT Madras, Chennai 600036, Tamil Nadu, India
Corresponding author: son.tran@deakin.edu.au,
Contributing author: seng.loke@deakin.edu.au, anil@ee.iitm.ac.in, namnp.0510@gmail.com

Abstract

High-order binary optimisation (HOBO) problems, defined as the minimisation of functions involving interactions among three or more binary variables, are prevalent in machine learning and quantum computing. The minimisation of these high-order energy functions remains computationally intractable without quadratization or reformulation. This work introduces FerQ, a universal and auxiliary-free reformulation of high-order energy functions based on Fermat quotient expansion. For any degree-d monomial over binary variables, FerQ provides an exact, closed-form representation as a rational linear combination of Fermat quotients evaluated on a scalar aggregate, with coefficients determined by a single matrix inversion. FerQ is evaluated against twelve existing energy function transformation methods on k-SAT and Max-k-SAT benchmarks from four databases, as well as on random p-spin glass and k-local Hamiltonian instances. FerQ achieves the highest satisfaction and weighted satisfaction rates across all benchmarks, while incurring the lowest CPU runtime. To enable deployment on quantum annealers, we derive FerQ-Bc, a qubit-efficient embedding scheme that maps the reformulated energies onto quadratic unconstrained binary optimisation (QUBO) form, achieving good ancilla efficiency in high-degree regimes.

1 Introduction

Backgrounds. Many real-world problems in machine learning (ML) can be written as the task of minimizing a polynomial energy function over binary variables [1, 2, 3, 4, 5]. This problem is also critical in quantum computing (QC), where energy functions are physically encoded as Hamiltonians on quantum hardware and minimized using quantum annealers [6] or variational quantum algorithms [7, 8]. In both research communities, the challenge deepens considerably when the energy function contains interactions among three or more variables simultaneously, which is a setting known as higher-order binary optimisation (HOBO). Such higher-order interactions arise naturally across a wide range of applications. In ML, they appear in structured prediction [9], Markov random fields [10, 11], neuro-symbolic reasoning [12, 13] and combinatorial reasoning tasks such as solving k-SAT and Max-k-SAT. In QC, they emerge in quantum-mechanical models such as p-spin glass models [14] or k-local Hamiltonians [15]. In practice, however, these HOBO formulations must be transformed into representations compatible with the constraints of physical quantum hardware [16, 17], which typically supports interactions up to second order (i.e., quadratic unconstrained binary optimisation or QUBO), in order to be solved. We survey existing methods for energy function reformulation in Section 2.

Contributions. We identify a connection between Fermat's Little Theorem and the representation of high-order Boolean monomials. Specifically, we show that the Fermat quotient evaluated on the scalar sum of binary variables in an interacting subset carries exact information about every higher-order interaction within that subset. This observation leads to the following two contributions:

1. FERQ: an auxiliary-free energy function reformulation (Sections 3.1 and 3.2). We prove that any high-order monomial over a set of binary variables is exactly representable as a rational linear combination of Fermat quotients, all evaluated on the same scalar aggregate of those variables (Theorem 3.2). The combination coefficients are determined in closed form by a single matrix inversion of fixed, precomputable size (Definition 1). The resulting method, FERQ, is universal (requiring no structural assumptions on the energy function) and introduces no auxiliary variables.
2. FERQ-BC: a qubit-efficient HOBO-to-QUBO encoder (Section 3.3). To deploy FERQ on quantum annealers, which admit only pairwise interactions, we propose the FERQ Binary-Counter (FERQ-BC) encoding. It replaces each high-order monomial with a single quadratic coupler by tracking the scalar aggregate of its constituent variables via a hybrid chain construction. For small subsets, a domain-wall chain is used; for larger ones, a binary-counter chain is used instead. In both cases, the number of ancilla qubits required grows only logarithmically with the size of the interacting subset (Corollary 3.5).

We evaluate both methods against 12 existing approaches on k-SAT, Max-k-SAT, random p-spin glass, and k-local Hamiltonian benchmarks (Section 4.1). FERQ achieves the highest satisfaction and weighted satisfaction rates across all benchmarks while incurring the lowest CPU runtime. In the high-degree regime, FERQ-BC reduces ancilla overhead by 57%–70% relative to the seven least efficient competing methods (Section 4.3).

2 Backgrounds and related works

A high-order energy function is denoted as

$$E(\mathbf{x}) = \sum_{i} h_i x_i + \sum_{i<j} J_{ij} x_i x_j + \sum_{i<j<k} K_{ijk} x_i x_j x_k + \cdots = \sum_{\mathcal{S}\subseteq[n]} J_{\mathcal{S}} \prod_{i\in\mathcal{S}} x_i, \tag{1}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)^\top \in \{0,1\}^n$ is a vector of n binary decision variables, $[n] := \{1, 2, \ldots, n\}$, and $J_{\mathcal{S}} \in \mathbb{R}$ is the real-valued interaction coefficient associated with the subset $\mathcal{S}$. The maximum cardinality of any interacting subset, $k := \max\{|\mathcal{S}| : J_{\mathcal{S}} \neq 0\}$, is called the order of the energy function; when $k \geq 3$, the problem is said to be genuinely high-order. The HOBO problem is then formulated as $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\{0,1\}^n} E(\mathbf{x})$. Because every monomial $\prod_{i\in\mathcal{S}} x_i$ is multilinear over $\{0,1\}^n$ (idempotency: $x_i^2 = x_i$), the energy function E is a multilinear pseudo-Boolean function $f : \{0,1\}^n \to \mathbb{R}$. The search space is of cardinality 2^n, rendering exhaustive search intractable for large n; moreover, HOBO subsumes QUBO ($k = 2$).

This central research programme concerns the systematic reformulation of intractable $E(\mathbf{x})$ into a tractable equivalent form. Existing methods are stratified into two main branches: (1) quadratisation and (2) reformulation. In former approaches, a pseudo-Boolean function $f : \{0,1\}^n \to \mathbb{R}$ of order $k \geq 3$ is tranformed into a quadratic pseudo-Boolean function $g : \{0,1\}^{n+a} \to \mathbb{R}$ of order $k' = 2$, through the introduction of $a \geq 0$ auxiliary binary variables $\mathbf{y} \in \{0,1\}^a$, such that $\min_{\mathbf{x}\in\{0,1\}^n} f(\mathbf{x}) = \min_{(\mathbf{x},\mathbf{y})\in\{0,1\}^{n+a}} g(\mathbf{x},\mathbf{y})$, with the additional requirement that for every $\mathbf{x} \in \{0,1\}^n$ there exists at least one $\mathbf{y}^*$ such that $g(\mathbf{x},\mathbf{y}^*) = f(\mathbf{x})$. In latter approaches, reformulation constitutes a broader class of transformations that re-express $E(\mathbf{x})$ into an alternative structured form that is not necessarily quadratic, but is more tractable for a specific solver architecture or algorithmic framework.

There are five main methods to transform HOBO to QUBO: ***(1)*** *Substitution-based* quadratisation [18] replaces higher-order terms with fresh auxiliary variables enforced via penalty constraints, which is the most elementary and historically foundational approach. ***(2)*** *Minimum-based* quadratisation [19] minimises the number of auxiliary variables introduced, establishing the theoretical ancilla lower bound; [20] further showed that submodular terms admit quadratisation with no auxiliary variables at all. ***(3)*** *Negative- and posiform-based* quadratisation [21] handles negatively signed interaction

terms, which resist direct penalty elimination and require dedicated sign-pattern gadgets. ***(4)*** *Algebraic* quadratisation [22] enforces product consistency between auxiliary and original variables via linearisation constraints such as Fortet inequalities or McCormick envelopes. ***(5)*** *Recursive and hierarchical* quadratisation [23, 24] reduces high-order terms iteratively via tree-structured decomposition, offering flexible ancilla allocation suited to sparse interaction hypergraphs. Across all five, the core trade-off between ancilla count and penalty magnitude governs the conditioning and solver compatibility of the resulting QUBO.

There are five main methods to reformulate HOBO into alternative structured forms. ***(1)*** *PUBO-to-Ising reformulation* [14] converts binary variables into spin variables, yielding an equivalent spin optimisation model directly compatible with quantum annealing hardware. ***(2)*** *Reformulation-Linearisation Technique (RLT)* [25] generates a hierarchy of progressively tighter linear programming relaxations by multiplying bound-factor products with problem constraints. ***(3)*** *Semidefinite reformulation via the Lasserre/SOS hierarchy* [26] re-expresses HOBO as a sequence of semidefinite programs over a positive moment matrix of increasing order, providing asymptotically exact bounds at exponentially growing computational cost. ***(4)*** *Factor graph and hypergraph reformulation* [27] encodes the energy function as a factor hypergraph, enabling approximate inference via generalized belief propagation and exact inference via junction-tree algorithms when treewidth is bounded. ***(5)*** *ILP embedding* [28] introduces continuous auxiliary variables with linearisation envelope constraints for each high-order term, reformulating HOBO as an integer linear program amenable to standard branch-and-cut solvers.

3 Method

Before proceeding, we present a running example to illustrate the main idea throughout the article.

Example 1. Given a quartic energy function

$$E(\mathbf{x}) = 5x_1x_2x_3x_4 - 3x_1x_2x_3 + x_1x_2 - 2x_3. \tag{2}$$

Let $a_{1234} = (x_1{+}x_2{+}x_3{+}x_4)$, $a_{123} = (x_1{+}x_2{+}x_3)$, $a_{12} = (x_1{+}x_2)$. We define $\delta_p(a) = (a - a^p)/p$, for p is a prime and compute the associate δ-values. Then, the interaction terms can be re-written in δ-notation as

$$\begin{aligned} x_1x_2 &= -\delta_2(a_{12}) \text{ by (32a)}; \qquad x_1x_2x_3 = \delta_2(a_{123}) - \tfrac{1}{2}\,\delta_3(a_{123}) \text{ by (32b) and (32c)} \\ x_1x_2x_3x_4 &= -\tfrac{1}{2}\,\delta_2(a_{1234}) + \tfrac{5}{16}\,\delta_3(a_{1234}) - \tfrac{1}{48}\,\delta_5(a_{1234}) \text{ by (32d)–(32f)} \end{aligned} \tag{3}$$

(2) is equivalent to

$$\tilde{E} = -\tfrac{5}{48}\,\delta_5(a_{1234}) + \tfrac{25}{16}\,\delta_3(a_{1234}) - \tfrac{5}{2}\,\delta_2(a_{1234}) + \tfrac{3}{2}\,\delta_3(a_{123}) - 3\,\delta_2(a_{123}) - \delta_2(a_{12}) - 2x_3. \tag{4}$$

Remark 1. We observe that any k-order interaction among n variables; i.e., $\prod_k x_k$ for $k < n$, can be written as a linear combination of δ-functions and linear terms of original variables.

3.1 Representations of high-order monomials

We formalize the observation following Example 1 as below:

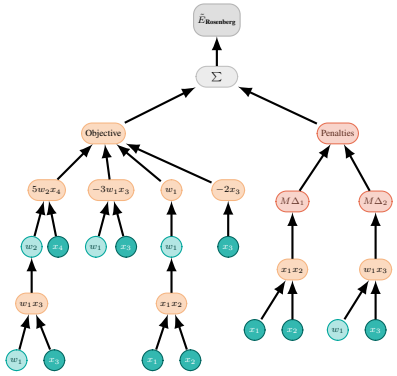
Lemma 3.1 (Fermat quotient expansion [30, 31, 32])**.** *Let $S \subseteq [n]$ with $|S| = m$, and let $x_i \in \{0,1\}$ for all $i \in S$. Define the integer-valued linear form $A_S = \sum_{i\in S} x_i; A_S \in \{0,1,\ldots,m\}$. For any prime p, the extended form of Fermat's Little Theorem*[1] *guarantees that $p \mid (A_S - A_S^p)$, so the* Fermat quotient

$$\delta_p(A_S) \;=\; \frac{A_S - A_S^p}{p} \;\in\; \mathbb{Z} \tag{5}$$

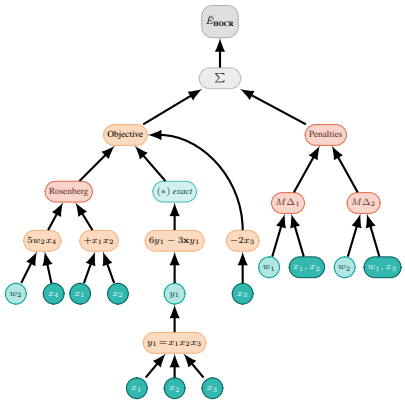
is a well-defined integer. Moreover, it admits the exact expansion

$$\delta_p(A_S) = -\frac{1}{p} \sum_{q=2}^{\min(p,\,m)} N(p,q)\; e_q(S), \quad e_q(S) = \sum_{\substack{T\subseteq S,\; i\in T \\ |T|=q}} \prod x_i; \tag{6}$$

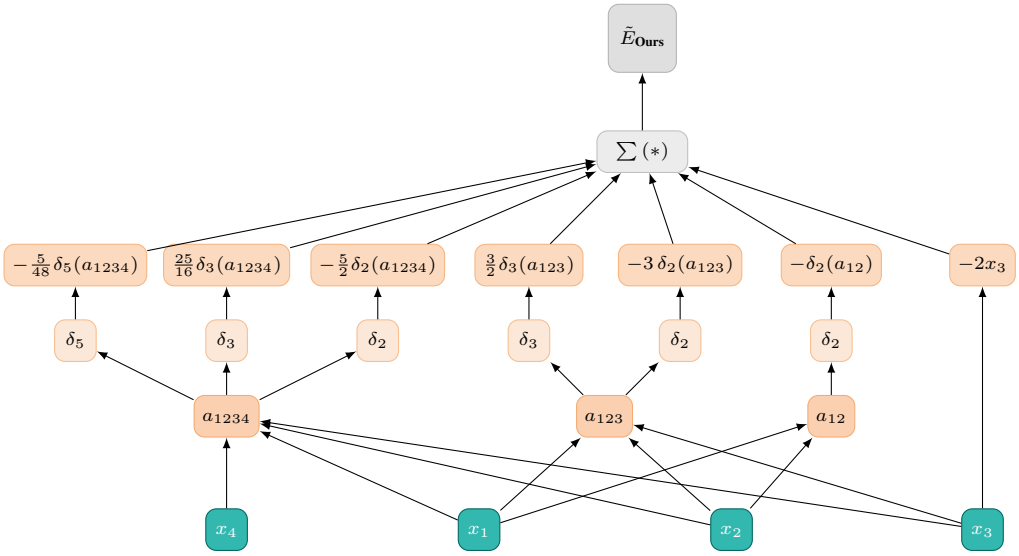
[1] Fermat's Little Theorem states $a^p \equiv a \pmod p$ for every prime p and every integer a; in particular, $p \mid (a - a^p)$ regardless of whether $\gcd(a,p) = 1$. The case $\gcd(a,p) > 1$ is often omitted in introductory statements but follows immediately since both sides vanish mod p when $p \mid a$.



(a) Rosenberg substitution: 6 total vars., 2 anc. vars.



(b) HOCR: 7 total vars., 3 anc. vars.



(c) Our proposed method (Energy function (10) using Theorem 3.2): 4 input vars., 0 anc. vars.

input vars. anc. vars. penalty

Figure 1: Computation graphs for **(a)** Rosenberg [18], **(b)** HOCR [20, 29], and **(c)** the proposed method applied to $E(\mathbf{x}) = 5x_1x_2x_3x_4 - 3x_1x_2x_3 + x_1x_2 - 2x_3$.

where $e_q(S)$ is the q-th elementary symmetric polynomial over the Boolean variables indexed by S, and $N(p,q) = q!\,S(p,q)$ is the number of surjections from $[p]$ onto a q-element set, with $S(p,q)$ the Stirling number of the second kind (Table F.1). In particular, $\delta_p(A_S)$ contains no degree-1 monomials.

Proof: see Appendix C.1 □

To recover each $e_q(S)$ and ultimately the degree-d monomial, we write Lemma 3.1 simultaneously for $d-1$ distinct primes, obtaining the linear system $M_d\,\mathbf{e} = -\boldsymbol{\delta}$. Each row of M_d corresponds to one prime and encodes the surjection coefficients $N(p_k,\, q+1)/p_k$ for degrees $q+1 = 2, \ldots, d$; the index shift from q to $q+1$ reflects the fact that $\mathbf{e}$ begins at e_2, since the degree-1 term cancels identically in Lemma 3.1.

Definition 1 (Coefficient Matrix M_d)**.** Let $d \geq 2$ and let $2 = p_1 < p_2 < \cdots < p_{d-1}$ be the first $d-1$ primes. Define $M_d \in \mathbb{Q}^{(d-1)\times(d-1)}$ by

$$[M_d]_{k,\,q} \;=\; \frac{N(p_k,\; q+1)}{p_k}, \qquad k,\, q \;\in\; \{1, \ldots, d-1\}, \tag{7}$$

where $N(p,q) = q!\,S(p,q)$ is the number of surjections from $[p]$ onto a q-element set, and $S(p,q)$ is the Stirling number of the second kind. We set $[M_d]_{k,q} = 0$ whenever $q+1 > \min(p_k,\, d)$, since

$S(p,q) = 0$ for $q > p$ and $e_q(S) = 0$ for $q > d$. Because M_d is lower triangular for $d \leq 4$ (diagonal entries $[M_d]_{k,k} = N(p_k, k+1)/p_k > 0$), its invertibility for small degrees follows immediately. For general d, invertibility is established in Lemma C.2.

The system $M_d\,\mathbf{e} = -\boldsymbol{\delta}$ encodes Lemma 3.1 simultaneously for $p_1, \ldots, p_{d-1}$, where

$$\mathbf{e} \;=\; \big(e_2(S),\; \ldots,\; e_d(S)\big)^\top, \qquad \boldsymbol{\delta} \;=\; \big(\delta_{p_1}(A_S),\; \ldots,\; \delta_{p_{d-1}}(A_S)\big)^\top. \tag{8}$$

The following theorem is the main result of this work:

Theorem 3.2. *Let $S \subseteq [n]$ with $|S| = d \geq 2$, and let $2 = p_1 < p_2 < \cdots < p_{d-1}$ be the first $d-1$ primes. Then the degree-d monomial $\prod_{i\in S} x_i$ admits the exact representation*

$$\prod_{i\in S} x_i \;=\; \sum_{k=1}^{d-1} \alpha_k^{(d)}\, \delta_{p_k}(A_S), \qquad A_S \;=\; \sum_{i\in S} x_i \tag{9}$$

where $\delta_{p_k}(A_S) = (A_S - A_S^{p_k})/p_k$ is the Fermat quotient defined in Lemma 3.1. The coefficient vector $\boldsymbol{\alpha}^{(d)}$ is the last row of $-M_d^{-1}$; i.e., row index $d-1$ under 1-based indexing, where M_d is the invertible matrix defined in Definition 1. We present computed $\boldsymbol{\alpha}^{(d)}$ in Table F.2.

Proof: see Appendix C.2 □

3.2 Representations of energy functions

Any HOBO objective in (1) decomposes naturally into three groups: linear terms, quadratic terms, and higher-order monomials of degree ≥ 3. Theorem 3.2 handles the third group exactly by replacing every degree-$|S|$ monomial $\prod_{i\in S} x_i$ with a linear combination of Fermat quotients $\delta_{p_k}(A_S)$. Each term evaluated at the scalar aggregate $A_S = \sum_{i\in S} x_i$. Specifically, $E(\mathbf{x})$ in (1) is exactly equivalent to (applying Theorem 3.2, Eq. 9 to every monomial of degree $|S| \geq 2$)

$$E(\mathbf{x}) \;=\; \sum_{i=1}^{n} w_{\{i\}}\, x_i \;+\; \sum_{\substack{S\subseteq[n]\\ 2\leq|S|\leq d}} w_S \underbrace{\sum_{k=1}^{|S|-1} \alpha_k^{(|S|)}\, \delta_{p_k}(A_S)}_{\prod_{i\in S} x_i} \tag{10}$$

where $A_S = \sum_{i\in S} x_i \in \{0, 1, \ldots, |S|\}$ is the integer-valued linear form over S, the Fermat quotient $\delta_{p_k}(A_S)$ is as defined in Lemma 3.1 (Eq. 5), and the coefficients $\alpha_k^{(|S|)}$ are derived following Definition 1 and Theorem 3.2. The end-to-end FERQ pipeline is presented in Algorithm 1.

3.3 Binary-counter HOBO-to-QUBO encoding

Although Theorem 3.2 re-expresses every high-order monomial as a linear combination of Fermat quotients evaluated on integer-valued aggregates, the resulting energy function retains its original degree and cannot be executed on quantum annealing hardware, which admits only quadratic interactions. We address this gap with the *FERQ Binary-Counter* (FERQ-BC) encoder: an exact HOBO-to-QUBO reduction in which each high-order monomial is replaced by a single new quadratic coupler, at the cost of a set of ancilla chain qubits whose number scales as $O(\log |T|)$ per sub-aggregate T (Corollary 3.5). Our proposed encoding scheme is based on three results: **(1)** we introduce a hybrid chain for aggregate A_S in Def. 2, for which a domain-wall chain for small sub-aggregates and a binary-counter chain otherwise; **(2)** Lemma 3.3 establishes that in both regimes the chain possesses a distinguished *output qubit* q^T satisfying $q^T = \prod_{i\in T} x_i$ on every valid chain state; and **(3)** Theorem 3.4 then collapses any degree-d monomial $\prod_{i\in S} x_i$ to the single quadratic coupler $x_{v_S} \cdot q^{T_S}$, where $T_S = S \setminus \{v_S\}$.

We begin with defining hybrid chain

Definition 2 (Hybrid chain)**.** Let $T \subseteq [n]$ with $|T| = m \geq 2$. The hybrid chain $\mathcal{H}_T$ encodes the aggregate $A_T := \sum_{i\in T} x_i$ is:

Case 1: domain-wall chain ($m \leq 5$). A tuple $(q_1^T, \ldots, q_m^T) \in \{0,1\}^m$ satisfying

$$q_1^T \geq q_2^T \geq \cdots \geq q_m^T, \qquad \sum_{j=1}^{m} q_j^T = A_T. \tag{11}$$

The valid states are exactly the $m+1$ configurations $q_j^T = \mathbf{1}[j \le k]$ for $k = A_T$. The output qubit is $q^T := q_m^T$.

Case 2: binary-counter chain ($m > 5$). Let $\ell := \lceil \log_2(m+1) \rceil$. Binary ancilla bits $\mathbf{r}^T = (r_0^T, \ldots, r_{\ell-1}^T)$ encode the integer

$$A_T \;=\; \sum_{j=0}^{\ell-1} 2^j \, r_j^T. \tag{12}$$

Here, we have $m = \sum_{j=0}^{\ell-1} m_j \, 2^j$ and define comparison bits

$$c_j^T \;:=\; \begin{cases} r_j^T & m_j = 1, \\ 1 - r_j^T & m_j = 0, \end{cases} \qquad j = 0, \ldots, \ell-1. \tag{13}$$

The intermediate ancillas is $a_1^T, \ldots, a_{\ell-2}^T$ and output qubit q^T via the conjunction tree

$$a_1^T = c_0^T c_1^T, \quad a_k^T = a_{k-1}^T \, c_k^T \;\; (k = 2, \ldots, \ell-2), \quad q^T = a_{\ell-2}^T \, c_{\ell-1}^T, \tag{14}$$

each enforced by the penalty as in [18]; i.e., $P(u,v,w) := uv - 2uw - 2vw + 3w \ge 0$, which vanishes iff $w = uv$.

We show that measuring the output qubit q^T is sufficient for computation of $\prod_{i \in T} x_i$ via the following lemma:

Lemma 3.3 (Terminal-qubit identity)**.** *Let $T \subseteq [n]$ with $|T| = m$, and let $(\mathbf{x}, \mathcal{H}_T)$ be a valid joint state. Then*

$$q^T \;=\; \mathbf{1}[A_T = m] \;=\; \prod_{i \in T} x_i. \tag{15}$$

Proof: see Appendix C.3 □

Then, any degree-d monomial can be collapsed to degree-2 monomial via the following theorem:

Theorem 3.4 (Chain linearisation)**.** *Let $S \subseteq [n]$ with $|S| = d \ge 3$, and fix any partition $S = T_S \cup \{v_S\}$ with $|T_S| = d-1$. Over every valid joint state of $\mathcal{H}_{T_S}$,*

$$\prod_{i \in S} x_i \;=\; x_{v_S} \cdot q^{T_S}. \tag{16}$$

The right-hand side is a degree-2 monomial, constituting a single QUBO coupler.

Proof: see Appendix C.4 □

The HOBO-to-QUBO encoding pipeline based on FERQ-BC is detailed in Algorithm 2. We analyze the correctness and ancilla cost ass follows:

Corollary 3.5. *Let $\widetilde{E}$ be the QUBO produced by Algorithm 2 with $\lambda > \sum_S |c_S|$. Then the following properties hold:*

- *(i)* $\min_{\mathbf{x}} \widetilde{E}(\mathbf{x}, \mathbf{q}^*) = \min_{\mathbf{x}} E(\mathbf{x})$*, where $\mathbf{q}^*$ is the unique ancilla state satisfying all validity constraints; equivalently, $\widetilde{E}(\mathbf{x}^*, \mathbf{q}^*) = E(\mathbf{x}^*)$ at any global minimiser $\mathbf{x}^*$.*
- *(ii) Each $S \in \mathcal{H}$ contributes exactly one quadratic coupler $x_{v_S} q^{T_S}$, independently of $|S|$.*
- *(iii) For each distinct sub-aggregate T with $m = |T|$, the total ancilla count is*

$$\nu(m) \;=\; \min\Big(m,\; 2\lceil \log_2(m+1) \rceil - 1\Big). \tag{17}$$

- *(iv) Every additional monomial $S' \in \mathcal{H}$ with $T_{S'} = T$ assigned to an existing chain $\mathcal{H}_T$ adds exactly one new coupler $x_{v_{S'}} q^T$ and* zero *new ancilla qubits.*

Proof: see Appendix C.5 □

We demonstrate FERQ-BC for HOBO-to-QUBO conversion in two worked examples in Appendix D.2 and present its pipeline in Algorithm 2.

Remark 2. The δ_p function in (5) serves the algebraic bridge from FERQ to FERQ-BC. Specifically, we showed that any high-order Boolean monomial is an exact deterministic function of its integer-valued aggregate A_S in Theorem 3.2; i.e, its linear expansion isolates monomial interactions with closed-form coefficients. As a result, tracking only the aggregate A_S suffices to recover target high-order monomial. This theoretical guarantee enables the hybrid chain in Def. 2, which is foundation of FERQ-BC.

4 Results

4.1 Baseline competitors

We compare FerQ against 12 existing quadratisation techniques for HOBO, including: Rosenberg-Sub [18], Ded-Red [33], Ntr-Kzfd [3, 34], Ntr-Abcg [21], Ntr-Abcg2 [21], Ntr-Gbp [5], Ptr-Ishikawa [29], Ptr-Kz [3], Ptr-Gbp [5], PairwiseCovers [21], Fgbz-Negative [35], Fgbz-Positive [35]. The prefix Ntr and Ptr denote negative and positive term reduction, respectively. The details of these methods are comprehensively discussed in a prior survey [23]. As a preliminary demonstration, we compare our proposed formulation with substitution-based methods, including Rosenberg [18] and HOCR/Ishikawa [20, 29], in Figure 1. The comparison of required #aux. vars for each method by #vars and polynomial orders are reported in Figure B.1.

4.2 Effectiveness analysis

4.2.1 k-SAT and Max-k-SAT

We use four databases including SATLIB[2], SAT Competition 2023-2025 (SATCOM23-25[3]) for k-SAT; and HAMLIB [36], MaxSAT Evaluation (MSE2024) [37] for Max-k-SAT. The SAT rate is computed as #satisfied clauses/#clauses. The approximation ratio (AR) is defined as $w_{\text{found}}/w_{\text{opt}}$, where unity indicates an optimal solution. The weighted satisfaction rate (WSR), $w_{\text{found}}/w_{\text{total}}$, measures the fraction of total clause weight satisfied; unlike the approximation ratio, it requires no knowledge of the optimum, but is strictly less than one on unsatisfiable instances.

Overall, FerQ achieves the highest SAT and weighted SAT rates across all four benchmarks while incurring the smallest CPU runtime, and it reaches the same performance when using unformulated HOBO.

On SATLIB (Figure 2, top panels), FerQ achieves the highest median satisfaction rate among all formulations, with a notably tight interquartile range, indicating robust and consistent performance. On the harder SATCOM23-25 (Figure 2, bottom panels), where most formulations exhibit considerable variance and several outliers fall below 0.85, FerQ maintains a high and stable satisfaction rate. In both settings, the runtime of FerQ on CPU is among the lowest across all competing formulations, remaining well below 10^5 ms at the median while many competitors frequently exceed 10^5–10^6 ms.

On MSE2024 (Figure 3), FerQ achieves the highest median weighted SAT rate, with most other formulations trailing noticeably and exhibiting heavier lower tails. The runtime advantage of FerQ is similarly preserved in this setting. On HAMLIB (Figure 4), FerQ consistently scores near 1.0 in both AR and WSR, and attains competitive zero average auxiliary variable by design.

4.2.2 Random p-spin glass model

The random p-spin glass model serves as a standard benchmark for evaluating HOBO, as it captures complex, non-linear interactions that are challenging to optimize. Three metrics are recorded per run: (1) average relative error $|E_{\text{truth}} - E^*|/|E^*|$ against the brute-force optimum, (2) average *is-optimal* rate, defined as the fraction of runs achieving exact recovery (absolute error $< 10^{-6}$), and (3) average ancilla variable count, measuring the qubit overhead of each energy reformulation method. As reported in Figure 5 (top panel), FerQ achieves a zero-mean relative error and an optimal recovery rate of 99.6% $(23/24$ configs.), placing it in a statistical tie with Fgbz-Positive (99.6%), Ntr-Abcg (99.2%), and the four methods that attain perfect 100% optimality, including Ntr-Gbp, Ptr-Kz, Ptr-Gbp, and Ntr-Kzfd. There is no statistically significant difference in solution quality against any of them (Mann–Whitney U test with all $p > 0.27$). FerQ is significantly better than the remaining seven methods ($p \leq 0.018$, up to $p = 1.6 \times 10^{-10}$). Its sole failure occurs at $(N = 14, p = 11)$, where its optimal rate drops to 90% with a considerably small mean relative error of 1.2×10^{-4}, a configuration on which Ntr-Gbp, Ptr-Kz, Ptr-Gbp, and Ntr-Kzfd all succeed. However, all of these methods use auxiliary variables. The details of this experiment is reported in Appendix E.1.

[2] https://www.cs.ubc.ca/~hoos/SATLIB/

[3] https://satcompetition.github.io/[2023,2024,2025]

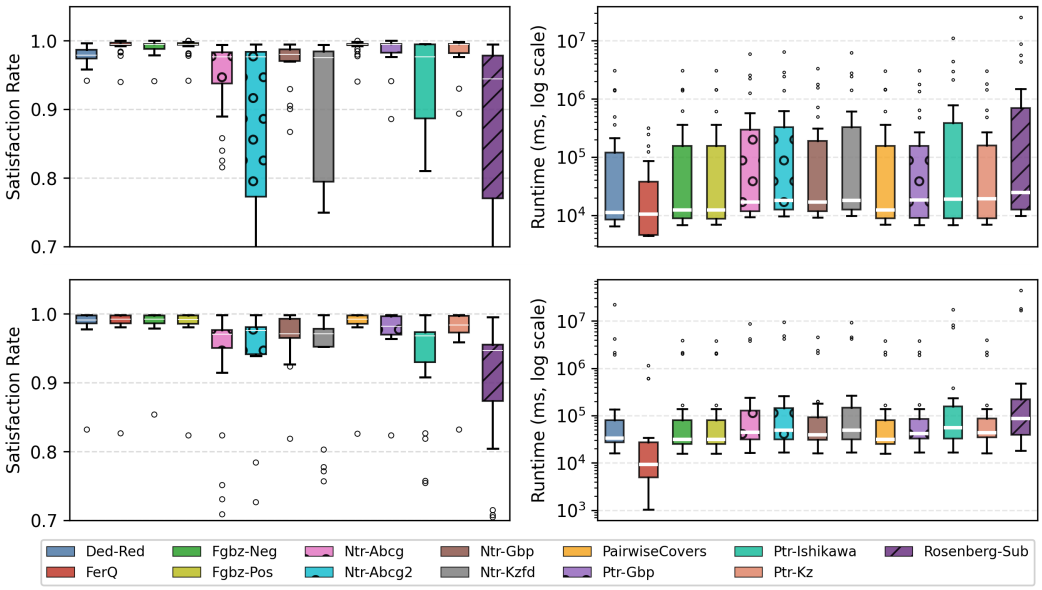


Figure 2: Experimental results in solving kSAT using SATLIB (#samples $= 24$) and SATCOM 23-25 (#samples $= 30$)

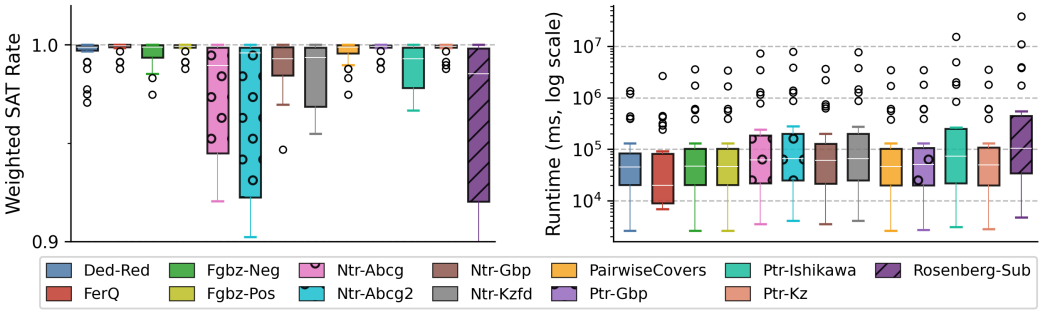


Figure 3: Experimental results in Max-kSAT using MSE2024 (#samples $= 26$)

4.2.3 k-local Hamiltonian

The k-local Hamiltonian problem provides a general framework for studying interactions involving up to k variables, extending beyond pairwise structures to more expressive higher-order systems. The comparison of FerQ against existing methods for this evaluation is reported in Figure 5 (bottom panel). FerQ achieves zero spectral error and 100% optimal rate across all configurations. Pairwise Mann-Whitney tests confirm FerQ is significantly better than Ded-Red, Fgbz-Negative, Ptr-Ishikawa, PairwiseCovers, and Rosenberg-Sub (all $p < 0.005$), and statistically indistinguishable from Ntr-Gbp ($p = 1.00$), Fgbz-Positive and Ptr-Kz (both with $p = 0.17$). FerQ reconstructs exact high-order energy functions without auxiliary variables, preserving the original search space and eliminating approximation error by construction, which is consistent with our theoretical expectations.

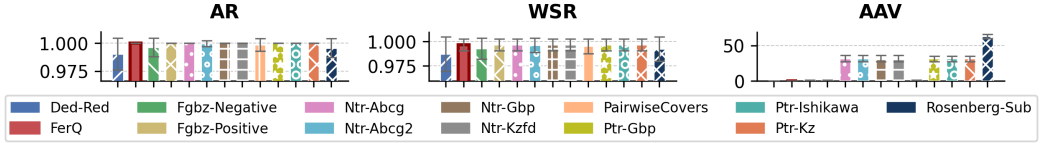


Figure 4: Experimental results in Max-kSAT using HAMLIB (#samples $= 30$)

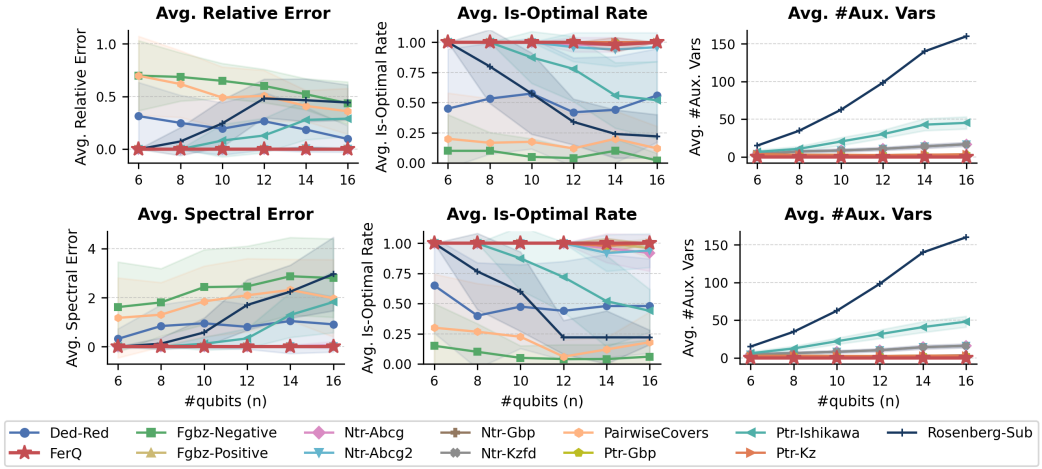


Figure 5: Experimental results in (top) random p-spin glass model and (bottom) k-local Hamiltonian

4.3 Ancilla-overhead comparison of FerQ-Bc with existing methods

As established in Section 3.3, FerQ inherently requires no auxiliary variables; however, its deployment via QUBO matrix encoding—yielding the FerQ-Bc variant—necessitates ancilla qubits. We quantify ancilla overhead as $(n_{\text{org}} + n_{\text{aux}})/n_{\text{org}}$, where lower values denote greater qubit efficiency relative to problem size. Figure B.2 reports results across 10,000 randomly generated energy functions with $n_{\text{org}} = 1000$ variables and interaction degrees spanning 3 to 43. In the low-degree regime ($\deg E \leq 8$), FerQ-Bc confers no advantage and occasionally incurs the highest overhead among evaluated methods. Conversely, for $\deg E > 8$, FerQ-Bc consistently achieves substantially fewer auxiliary variables, outperforming seven of the twelve benchmarked approaches. Its overhead remains marginally above that of Ntr-based encodings and Ptr-Ishikawa. Notably, in the extreme high-degree regime ($\deg E \in [31, 43]$), FerQ-Bc requires only 150 auxiliary variables, whereas the seven less efficient methods demand 350–500, corresponding to a 57%–70% reduction in ancilla count.

5 Discussion and conclusion

GPU acceleration performance of FerQ. We develop a CUDA-based implementation of FerQ (FerQ-Gpu) and compare with CPU-based implementation (FerQ-Cpu). FerQ-Gpu is slower than FerQ-Cpu when parallelism is low, and only starts to help when #reads ≥ 256 (Figure B.3). At #reads $= 50$, the GPU is always slower for all problem sizes, giving only $0.2\times$-$0.4\times$ speedup. This happens because the overhead from launching kernels and moving data is larger than the actual computation when there are too few threads. As #reads increases, the speedup also increases steadily for all problem sizes, reaching $5.4\times$-$12.0\times$ at #reads $= 2048$. However, at high read counts, the speedup becomes smaller for larger problems (Figure B.4). For example, small problems (#var $= 50$) get the highest speedup ($12.0\times$), while large ones (#var $= 1000$) get the lowest ($5.4\times$). This is because larger problems require more memory and more work per thread, which slows down the GPU. FerQ-Cpu increases quickly with both #var and #reads (up to ≈ 2000 ms at #var $= 1000$, #reads $= 2048$). In contrast, FerQ-Gpu grows much more slowly (about 300 ms for the same case), showing that it benefits from handling many reads at once. Overall, the GPU is useful only when #reads is large, and its advantage becomes smaller as the problem size increases. This experiment is run on NVIDIA T4 GPU with GPU RAM of 16 GB.

Implications for ML. Methods discussed at the beginning of Section 4.1 have been applied to machine learning tasks; therefore, our proposed method is also applicable to these settings. While its effectiveness may vary by task and requires further evaluation, results in Section 4.1 show that it achieves competitive performance without introducing additional auxiliary variables. Here, we outline several future research directions that could benefit the machine learning community. In computer vision, related methods have been successfully applied to tasks such as stereo correspondence, multi-camera scene reconstruction, and voxel occupancy estimation [1, 2, 3].Our method can be

extended to support image processing tasks, including segmentation, restoration, and denoising, often leveraging expressive priors such as Fields of Experts (FoE) [4]. Furthermore, structured labeling problems, highlight the potential of these approaches for real-world perception systems [5].

Comparison with zero-ancilla quadratisation methods. FERQ shares with zero-auxiliary methods; including DED-RED [33], ELC reduction [38], Gröbner bases [39], and split reduction [40], the critical advantage of introducing no ancilla variables. However, those methods are either problem-specific or combinatorially expensive [23]: DED-RED and ELC reduction require deductions or excludable configurations that may not exist or are exponentially hard to find; Gröbner bases only preserve roots and are thus algebraically restricted for global minimisation; and split reduction doubles the number of subproblems with each application. By contrast, FERQ is a universal, deterministic, and closed-form transformation requiring no structural preconditions, yielding an exactly equivalent energy function that preserves the full objective landscape without penalty augmentation or exponential enumeration.

Tractability of FERQ. Figure 1(c) demonstrates the tractability of FERQ by revealing how the combinatorial intractability of high-order monomials is resolved through scalar aggregation. Specifically, rather than operating directly on the exponentially large joint configuration space $\{0,1\}^4$, the computation graph shows that every high-order interaction is exactly captured through integer-valued linear aggregates a_{1234}, a_{123}, and a_{12}, each taking values in a linear range $\{0,\dots,m\}$ rather than an exponential one. These scalar aggregates are then passed through Fermat quotient operators δ_2, δ_3, and δ_5, which encode all high-order interaction information. Critically, the figure confirms that this tractability is achieved without expanding the search space (no ancilla variable nodes appear) meaning the original 2^4 feasible configurations are preserved exactly, with no spurious local minima introduced by penalty augmentation. The intractable high-order energy function is thus rendered tractable through a deterministic, closed-form transformation of polynomial cost.

Applicability criteria for FERQ-BC. From the result presented in Section 4.3, FERQ-BC is recommended when encoding high-order energy functions ($\deg E \gtrsim 8$) prevalent in expressive machine learning models; such as high-order Markov random fields, tensorized feature interactions, and constrained structured prediction, where preserving interaction arity is essential for model fidelity. In quantum computing, it is particularly suited for embedding multi-body Hamiltonians from quantum chemistry and lattice gauge theories, where its ancilla-efficient structure directly alleviates qubit scarcity and coherence constraints on NISQ devices. Conversely, FERQ-BC should be deferred for low-degree or pairwise-dominant formulations, including standard QUBO reductions, binary classification with linear kernels, and graph-cut problems, where established pairwise encodings offer equal or superior qubit efficiency. Thus, FERQ-BC serves as a specialized embedding primitive for high-interaction workloads, bridging algorithmic expressivity and hardware feasibility in near-term quantum-enhanced learning.

Limitations. Three limitations bound the current scope of FERQ. First, the degree ceiling of 43 is not fundamental to the theory but is an arithmetic constraint: the surjection coefficients $N(p_k, q)$ grow super-exponentially with p_k, eventually exceeding 64-bit integer precision. Arbitrary-precision arithmetic would lift this ceiling at the cost of runtime. Second, coefficient computation requires inverting M_d, while this is a one-time offline cost per degree, it becomes a bottleneck for very high-degree instances and may benefit from symbolic precomputation or structured factorisation. Third, FERQ reformulates but does not reduce degree: the output energy retains the same interaction order as the input, so direct execution on quantum annealers requires the additional FERQ-BC encoding step.

Conclusion. FERQ establishes a fundamentally new bridge between classical number theory and combinatorial optimisation, demonstrating that Fermat's Little Theorem enables an exact, universal, and ancilla-free pathway through the intractability of high-order binary energy minimisation. Looking forward, co-designing FERQ with next-generation quantum hardware represents the most immediate frontier, where its constant per-term coupling cost may prove decisive as annealer connectivity and qubit counts scale. More broadly, whether the Fermat quotient machinery extends gracefully to non-Boolean discrete domains or admits continuous relaxations compatible with gradient-based solvers remains an open question that could substantially widen the reach of the proposed framework.

Acknowledgement

All authors declare no competing interests. The work was supported by the Deakin–IIT Madras Joint Project (No. IITM-250601-157) and Deakin University (No. GR06084).

References

[1] Tomaso Poggio, Vincent Torre, and Christof Koch. Computational vision and regularization theory. *Readings in computer vision*, pages 638–643, 1987.

[2] Dan Snow, Paul Viola, Ramin Zabih, et al. Exact voxel occupancy with graph cuts. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–345. IEEE Computer Society; 1999, 2000.

[3] Vladimir Kolmogorov and Ramin Zabin. What energy functions can be minimized via graph cuts? *IEEE transactions on pattern analysis and machine intelligence*, 26(2):147–159, 2004.

[4] Asem M Ali, Aly A Farag, and Georgy L Gimel'Farb. Optimizing binary mrfs with higher order cliques. In *European Conference on Computer Vision*, pages 98–111. Springer, 2008.

[5] Andrew C Gallagher, Dhruv Batra, and Devi Parikh. Inference for order reduction in markov random fields. In *CVPR 2011*, pages 1857–1864. IEEE, 2011.

[6] Mark W Johnson, Mohammad HS Amin, Suzanne Gildert, Trevor Lanting, Firas Hamze, Neil Dickson, Richard Harris, Andrew J Berkley, Jan Johansson, Paul Bunyk, et al. Quantum annealing with manufactured spins. *Nature*, 473(7346):194–198, 2011.

[7] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[8] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):4213, 2014.

[9] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin markov networks. *Advances in neural information processing systems*, 16, 2003.

[10] Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

[11] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[12] Son N Tran and Artur d'Avila Garcez. Neurosymbolic reasoning and learning with restricted boltzmann machines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6558–6565, 2023.

[13] Son Tran, Edjard Mota, and Artur d'Avila Garcez. Reasoning in neurosymbolic ai. *Neurosymbolic AI: Foundations and Applications*, pages 15–68, 2026.

[14] Andrew Lucas. Ising formulations of many np problems. *Frontiers in physics*, 2:74887, 2014.

[15] Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. *Siam journal on computing*, 35(5):1070–1097, 2006.

[16] Kelly Boothby, Paul Bunyk, Jack Raymond, and Aidan Roy. Next-generation topology of d-wave quantum processors. *arXiv preprint arXiv:2003.00133*, 2020.

[17] Ana Palacios, Artur Garcia-Saez, Bruno Juliá-Díaz, and Marta P Estarellas. Scalable 2-local architecture for quantum annealing of ising models with arbitrary dimensions. *Physical Review Applied*, 23(5):054070, 2025.

[18] Ivo G Rosenberg. Reduction of bivalent maximization to the quadratic case. 1975.

[19] Endre Boros and Peter L Hammer. Pseudo-boolean optimization. *Discrete applied mathematics*, 123(1-3):155–225, 2002.

[20] Hiroshi Ishikawa. Higher-order clique reduction in binary graph cut. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2993–3000. IEEE, 2009.

[21] Martin Anthony, Endre Boros, Yves Crama, and Aritanan Gruber. Quadratic reformulations of nonlinear binary optimization problems. *Mathematical Programming*, 162(1):115–144, 2017.

[22] Peter L Hammer and Sergiu Rudeanu. *Boolean methods in operations research and related areas*. Springer Science & Business Media, 2012.

[23] Nike Dattani. Quadratization in discrete optimization and quantum mechanics. *arXiv preprint arXiv:1901.04405*, 2019.

[24] Nike Dattani and Nick Chancellor. Embedding quadratization gadgets on chimera and pegasus graphs. *arXiv preprint arXiv:1901.07676*, 2019.

[25] Hanif D Sherali and Warren P Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990.

[26] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001.

[27] Jonathan S Yedidia, William T Freeman, Yair Weiss, et al. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8(236-239):0018–9448, 2003.

[28] Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

[29] Hiroshi Ishikawa. Transformation of general binary mrf minimization to the first-order case. *IEEE transactions on pattern analysis and machine intelligence*, 33(6):1234–1249, 2010.

[30] Alexandru Buium. Differential characters of abelian varieties over p-adic fields. *Inventiones mathematicae*, 122(1):309–340, 1995.

[31] Victor H Moll. Numbers and functions. *AMS, Providence*, 2012.

[32] Alexandru Buium and Santiago R Simanca. Arithmetic laplacians. *Advances in Mathematics*, 220(1):246–277, 2009.

[33] Richard Tanburn, Emile Okada, and Nike Dattani. Reducing multi-qubit interactions in adiabatic quantum computation without adding auxiliary qubits. part 1: The" deduc-reduc" method and its application to quantum factorization of numbers. *arXiv preprint arXiv:1508.04816*, 2015.

[34] Daniel Freedman and Petros Drineas. Energy minimization via graph cuts: Settling what is possible. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 939–946. IEEE, 2005.

[35] Alexander Fix, Aritanan Gruber, Endre Boros, and Ramin Zabih. A graph cut algorithm for higher-order markov random fields. In *2011 International Conference on Computer Vision*, pages 1020–1027. IEEE, 2011.

[36] Nicolas PD Sawaya, Daniel Marti-Dafcik, Yang Ho, Daniel P Tabor, David E Bernal Neira, Alicia B Magann, Shavindra Premaratne, Pradeep Dubey, Anne Matsuura, Nathan Bishop, et al. Hamlib: A library of hamiltonians for benchmarking quantum algorithms and hardware. *Quantum*, 8:1559, 2024.

[37] Maxsat evaluation 2024, 2024. 19th MaxSAT Evaluation, accessed 2026-04-08.

[38] Hiroshi Ishikawa. Higher-order clique reduction without auxiliary variables. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1362–1369, 2014.

[39] Raouf Dridi and Hedayat Alghassi. Prime factorization using quantum annealing and computational algebraic geometry. *Scientific reports*, 7(1):43048, 2017.

[40] Emile Okada, Richard Tanburn, and Nikesh S Dattani. Reducing multi-qubit interactions in adiabatic quantum computation without adding auxiliary qubits. part 2: The" split-reduc" method and its application to quantum determination of ramsey numbers. *arXiv preprint arXiv:1508.07190*, 2015.

[41] Richard P Stanley. Enumerative combinatorics volume 1 second edition. *Cambridge studies in advanced mathematics*, 2011.

[42] Richard P. Stanley. *Enumerative Combinatorics, Volume 2*, volume 62 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2001. First published 1999.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We support our claims in the abstract theoretically through Section 3 and empirically through Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We address the limitation of the current work in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide proofs of main theoretical results in Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed experimental design in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All datasets used in this work is publicly available. We also submitted the implementation Python code.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide experimental design, data generation and hyperparameter setting in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We perform statistical tests in Section 4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the computation results the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: Our work does not have societal impacts to extend of our knowledge. However, we discuss the impacts on machine learning and quantum computing research communities throughout the work.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A].

Justification: The research does not impose any risk to extend of our knowledge.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets are appropriately cited to their original sources, and all implementation code is developed independently by us.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A].

    Justification: The present work does not released any new asset.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A].

Justification: LLMs were employed solely for writing, editing, and formatting tasks, and had no influence on the core methodology.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

A Algorithms

Algorithm 1: FERQ: HOBO Monomial Reformulation via Fermat Quotients

Input: HOBO objective $E(\mathbf{x}) = \sum_{S \subseteq [n]} w_S \prod_{i \in S} x_i$, with $x_i \in \{0, 1\}$

Output: Equivalent energy $\tilde{E}(\mathbf{x})$ expressed solely in linear and Fermat-quotient terms (Eq. 10)

▷ **Partition monomials by degree**

$\mathcal{L} \leftarrow \{S : |S| \leq 1,\ w_S \neq 0\}$, $\mathcal{Q} \leftarrow \{S : |S| = 2,\ w_S \neq 0\}$, $\mathcal{H} \leftarrow \{S : |S| \geq 3,\ w_S \neq 0\}$

▷ **Precompute coefficient matrices** (Def. 1)

$d_{\max} \leftarrow \max_{S \in \mathcal{H}} |S|$

for $d \leftarrow 2$ **to** $d_{\max}$ **do**

  let $p_1 < p_2 < \cdots < p_{d-1}$ be the first $d-1$ primes

  build $M_d \in \mathbb{Q}^{(d-1)\times(d-1)}$ with $[M_d]_{k,q} = N(p_k,\, q+1)/p_k$

  ▷ $N(p, q) = q!\, S(p, q)$; set entry to 0 when $q + 1 > \min(p_k, d)$

  $\boldsymbol{\alpha}^{(d)} \leftarrow -\left(M_d^{-1}\right)_{d-1,\,\cdot}$

  ▷ last row of $-M_d^{-1}$ gives monomial coefficients (Thm. 3.2)

▷ **Linearize quadratic terms via Fermat quotient** (Lem. 3.1, $d = 2$)

foreach $S \in \mathcal{Q}$ **do**

  $A_S \leftarrow \sum_{i \in S} x_i$

  replace $w_S \prod_{i \in S} x_i \ \leftarrow\ w_S\, \alpha_1^{(2)}\, \delta_{p_1}(A_S)$

▷ **Linearize higher-order monomials** (Thm. 3.2)

foreach $S \in \mathcal{H}$ **do**

  $d \leftarrow |S|$

  $A_S \leftarrow \sum_{i \in S} x_i$

  for $k \leftarrow 1$ **to** $d-1$ **do**

    $\delta_{p_k}(A_S) \leftarrow \dfrac{A_S - A_S^{p_k}}{p_k}$

    ▷ Fermat quotient; integer-valued by Lem. 3.1

  replace $w_S \prod_{i \in S} x_i \ \leftarrow\ w_S \sum_{k=1}^{d-1} \alpha_k^{(d)}\, \delta_{p_k}(A_S)$

▷ **Assemble equivalent energy** (Eq. 10)

$$\tilde{E}(\mathbf{x}) \ \leftarrow\ \sum_{S \in \mathcal{L}} w_S\, x_i \ +\ \sum_{S \in \mathcal{Q} \cup \mathcal{H}} w_S \sum_{k=1}^{|S|-1} \alpha_k^{(|S|)}\, \delta_{p_k}(A_S)$$

return $\tilde{E}$

Algorithm 2: FERQ-BC: HOBO-to-QUBO Encoder

Input: HOBO objective $E(\mathbf{x}) = \sum_{S \subseteq [n]} c_S \prod_{i \in S} x_i$, penalty $\lambda > \sum_S |c_S|$

Output: QUBO $\widetilde{E}(\mathbf{x}, \mathbf{q})$ with $\min_{\mathbf{x}} \widetilde{E} = \min_{\mathbf{x}} E$

▷ **Partition monomials**

$\mathcal{H} \leftarrow \{S : |S| \geq 3,\ c_S \neq 0\}$ $\mathcal{L} \leftarrow \{S : |S| \leq 2,\ c_S \neq 0\}$

▷ **Sub-aggregate selection**

foreach $S \in \mathcal{H}$ **do**

  choose $T_S \subsetneq S$ with $|T_S| = |S| - 1$

  $v_S \leftarrow S \setminus T_S$

Group monomials by shared T_S to maximize ancilla reuse

▷ **Construct hybrid chains (Def. 2)**

foreach *distinct* $T \in \{T_S : S \in \mathcal{H}\}$ **do**

  $m \leftarrow |T|$

  if $m \leq 5$ **then**

    add penalty: $P_{\mathcal{H}_T} \leftarrow \sum_{k=1}^{m-1} q_{k+1}^T (1 - q_k^T) + \left(\sum_{k=1}^{m} q_k^T - A_T\right)^2$

  else

    add penalty: $P_{\mathcal{H}_T} \leftarrow \left(\sum_{j=0}^{\ell-1} 2^j r_j^T - A_T\right)^2 + \sum_k \text{Penalty}$

  add $\lambda \cdot P_{\mathcal{H}_T}$ to total penalty

▷ **Linearize higher-order terms** (Theorem 3.4)

foreach $S \in \mathcal{H}$ **do**

  replace $c_S \prod_{i \in S} x_i \leftarrow c_S\, x_{v_S} \cdot q^{T_S}$

▷ **Sub-product sharing**

foreach T *that appears as a monomial and sub-aggregate* **do**

  replace $c_T \prod_{i \in T} x_i \leftarrow c_T\, q^T$

▷ **Assemble QUBO**

$\widetilde{E}(\mathbf{x}, \mathbf{q}) \leftarrow \sum_{S \in \mathcal{L}} c_S \prod_{i \in S} x_i + \sum_{S \in \mathcal{H}} c_S\, x_{v_S}\, q^{T_S} + \lambda \sum_T P_{\mathcal{H}_T}$

return $\widetilde{E}$

B Supplemental Figures

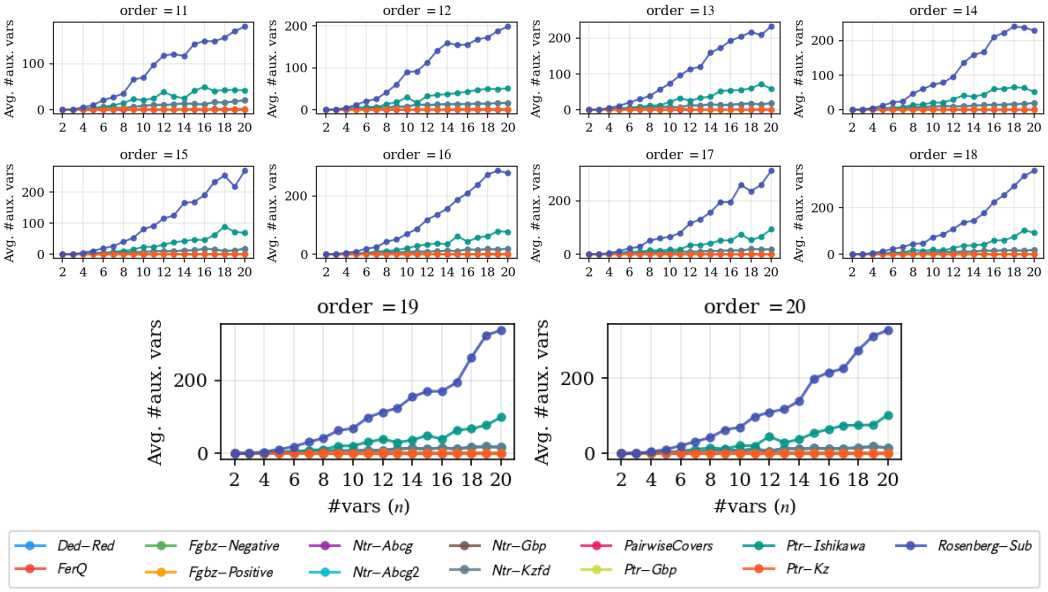


Figure B.1: Comparison of the average #aux. vars required by FerQ against reformulation methods for high-order energy functions. The number of original variables ranges from 2 to 20, with interaction orders from 11 to 20. Results are averaged over five random instances per order. Rosenberg-Sub [18] exhibits the highest overhead ($n_{\text{aux}} = 325.7$), followed by Ptr-Ishikawa [29] ($n_{\text{aux}} = 102.3$). In contrast, Ded-Red [33] and the proposed FerQ introduce no auxiliary variables.

C Proofs of theorems

Throughout, $[n] := \{1, 2, \ldots, n\}$ for any positive integer n. Fix a ground set $S \subseteq [n]$ with $|S| = m$. Every variable x_i takes values in $\{0, 1\}$, so the vector $\mathbf{x} = (x_i)_{i \in S}$ is Boolean. The *support* of $\mathbf{x}$ is $\operatorname{supp}(\mathbf{x}) := \{i \in S : x_i = 1\}$. The following definitions and theorem will be used for the further proofs:

Definition 3 (Elementary symmetric polynomials.)**.** For $q \geq 0$ define

$$e_q(S) \;:=\; \sum_{\substack{T \subseteq S \\ |T| = q}} \prod_{i \in T} x_i, \qquad e_0(S) := 1, \quad e_q(S) := 0 \text{ for } q > m. \tag{18}$$

Remark 3. Because every $x_i \in \{0, 1\}$ satisfies $x_i^k = x_i$ for all $k \geq 1$, the product $\prod_{i \in T} x_i$ equals 1 if $T \subseteq \operatorname{supp}(\mathbf{x})$ and 0 otherwise. Hence

$$e_q(S) \;=\; \binom{A_S}{q}, \qquad A_S \;:=\; \sum_{i \in S} x_i \;\in\; \{0, 1, \ldots, m\}, \tag{19}$$

where we use the convention $\binom{a}{q} = 0$ for $a < q$. Equation (19) is fundamental: it reduces every $e_q(S)$ to a function of the *scalar* A_S.

Definition 4 (Stirling numbers and surjections)**.** The *Stirling number of the second kind* $S(p, q)$ counts the number of set partitions of $[p]$ into exactly q non-empty blocks. It satisfies $S(p, q) = 0$ for $q > p$ or $q < 1$, $S(p, 1) = S(p, p) = 1$ for $p \geq 1$, and the recurrence $S(p, q) = q\, S(p-1, q) + S(p-1, q-1)$. The number of *surjective* functions from a p-element set to a q-element set is

$$N(p, q) \;:=\; q!\, S(p, q). \tag{20}$$

Equivalently, by inclusion-exclusion,

$$N(p, q) \;=\; \sum_{j=0}^{q} (-1)^{q-j} \binom{q}{j} j^p. \tag{21}$$

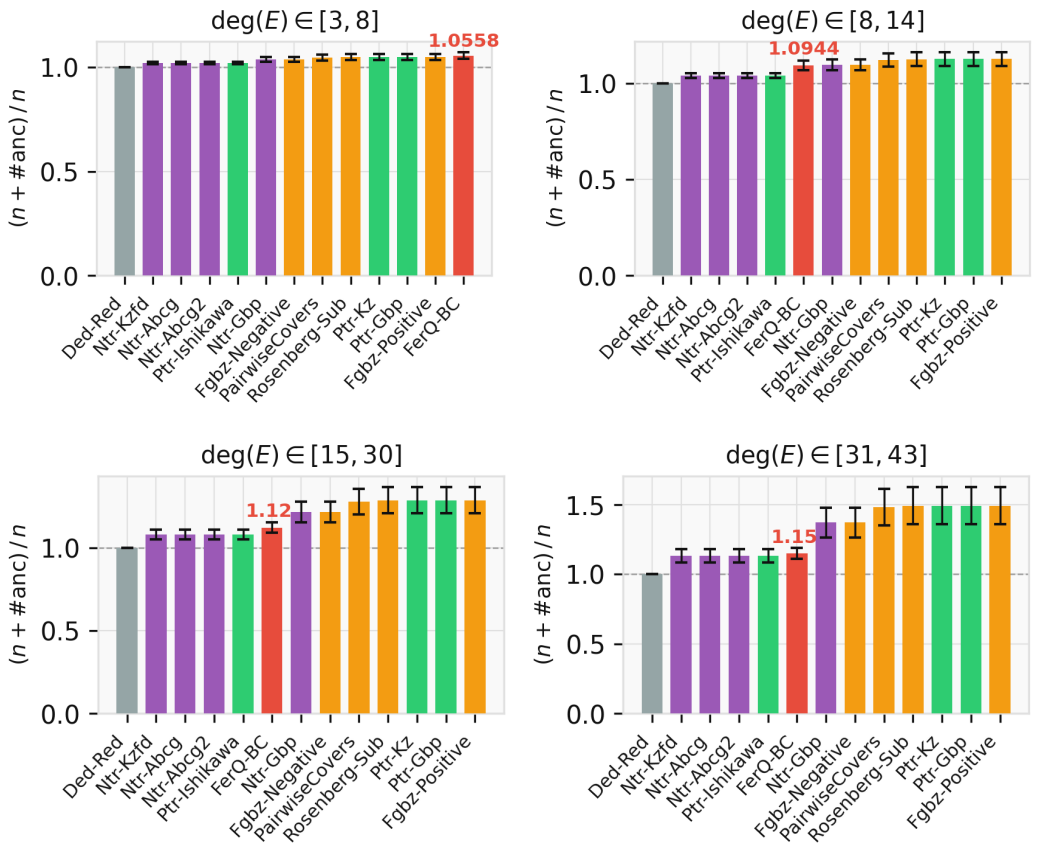


Figure B.2: Ancilla-overhead comparison of FerQ-Bc with existing methods

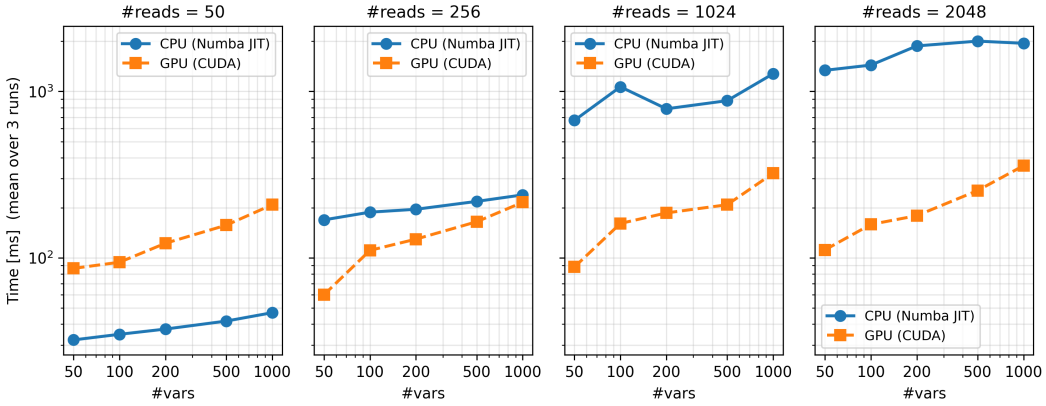


Figure B.3: Runtime comparison of FerQ using CPU and GPU backends

A special value is $N(p,1) = 1$ for all $p \geq 1$; $N(p,q) = 0$ for $q > p$.

Theorem C.1 (Falling-factorial / Stirling expansion of a power)**.** *For any integer a and positive integer p,*

$$a^p = \sum_{q=1}^{p} N(p,q) \binom{a}{q}, \tag{22}$$

as a polynomial identity in a (where $\binom{a}{q}$ is the falling-factorial polynomial $a(a-1)\cdots(a-q+1)/q!$). Equation (22) is the standard representation of a power in the falling-factorial basis; it can be verified by comparing both sides as polynomials in a using the definition of $S(p,q)$

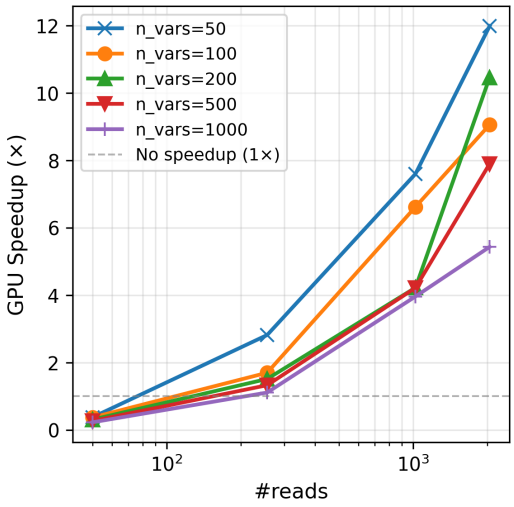


Figure B.4: Speedup of FERQ using GPU-CUDA backend

Proof. See page 81, Table 1, entry 3 and page 543 in [41]. □

C.1 Proof of Lemma 3.1

We prove the three claims in the Lemma sequentially: **(i)** well-definedness, and **(ii)** the expansion formula.

Proof. **(i) Well-definedness.** By Fermat's Little Theorem, for every integer a and every prime p, $a^p \equiv a \pmod{p}$. Since $A_S \in \mathbb{Z}$,

$$A_S^p \equiv A_S \pmod{p}, \quad \text{i.e.,} \quad p \mid (A_S - A_S^p).$$

Hence $\delta_p(A_S) = (A_S - A_S^p)/p \in \mathbb{Z}$.

(ii) The expansion formula. We compute A_S^p via the multinomial theorem and simplify using the Boolean constraint $x_i^k = x_i$ $(k \geq 1)$.

First, we write $A_S = \sum_{i \in S} x_i$. By the multinomial theorem,

$$A_S^p = \left(\sum_{i \in S} x_i\right)^p = \sum_{\substack{\mathbf{k}=(k_i)_{i\in S} \in \mathbb{Z}_{\geq 0}^m \\ \sum_{i\in S} k_i = p}} \frac{p!}{\prod_{i\in S} k_i!} \prod_{i \in S} x_i^{k_i}.$$

Because $x_i \in \{0,1\}$, we have $x_i^{k_i} = x_i$ for $k_i \geq 1$ and $x_i^0 = 1$. For each multiindex $\mathbf{k}$, define its *support*

$$T(\mathbf{k}) := \{i \in S : k_i \geq 1\}.$$

Then $\prod_{i\in S} x_i^{k_i} = \prod_{i \in T(\mathbf{k})} x_i$. Group all indices that share the same support $T \subseteq S$:

$$A_S^p = \sum_{\emptyset \neq T \subseteq S} \left(\sum_{\substack{(k_i)_{i\in T}:\, k_i \geq 1 \\ \sum_{i\in T} k_i = p}} \frac{p!}{\prod_{i\in T} k_i!} \right) \prod_{i \in T} x_i. \tag{23}$$

Fix T with $|T| = q$. The inner sum in (23)

$$C(p,T) := \sum_{\substack{(k_i)_{i\in T}:\, k_i \geq 1 \\ \sum k_i = p}} \frac{p!}{\prod_{i\in T} k_i!}$$

counts the number of surjective functions $f : [p] \to T$, as follows. Each surjection f induces the composition $k_i = |f^{-1}(i)| \geq 1$ with $\sum k_i = p$; the number of functions f yielding a given composition (k_i) is the multinomial coefficient $p!/\prod k_i!$. Summing over all compositions with $k_i \geq 1$ and $\sum k_i = p$ therefore counts all surjections from $[p]$ to T. Since $|T| = q$, this count depends only on q and equals $N(p,q) = q!\, S(p,q)$ (with $N(p,q) = 0$ for $q > p$, as no surjection onto a larger set exists). Thus $C(p,T) = N(p,|T|)$.

Substituting back into (23) and grouping by $q = |T|$:

$$A_S^p \;=\; \sum_{q=1}^{\min(p,m)} N(p,q) \underbrace{\sum_{\substack{T\subseteq S\\ |T|=q}} \prod_{i\in T} x_i}_{e_q(S)}. \tag{24}$$

(The upper limit is $\min(p,m)$ because $N(p,q) = 0$ for $q > p$ and $e_q(S) = 0$ for $q > m = |S|$.)

Since $N(p,1) = 1!\, S(p,1) = 1$ for all $p \geq 1$ and $e_1(S) = A_S$:

$$A_S^p \;=\; A_S \;+\; \sum_{q=2}^{\min(p,m)} N(p,q)\, e_q(S).$$

Subtracting and dividing by p:

$$\delta_p(A_S) \;=\; \frac{A_S - A_S^p}{p} \;=\; -\frac{1}{p} \sum_{q=2}^{\min(p,m)} N(p,q)\, e_q(S),$$

which is the claimed expansion in Lemma 3.1. □

C.2 Proof of Theorem 3.2

Fix $S \subseteq [n]$ with $|S| = d \geq 2$ and let $2 = p_1 < p_2 < \cdots < p_{d-1}$ be the first $d-1$ primes. We will give the proof in four steps: **(1)** setting up the linear system, **(2)** computing the invertibility of M_d, **(3)** solving for $\mathbf{e}$, and **(4)** extracting the degree-d monomial.

Proof. **(1)** Apply Lemma 3.1 to each prime p_k ($k = 1, \ldots, d-1$). Since $|S| = d$, $e_q(S) = 0$ for $q > d$, so the upper limit simplifies:

$$\delta_{p_k}(A_S) \;=\; -\frac{1}{p_k} \sum_{q=2}^{\min(p_k,\, d)} N(p_k,q)\, e_q(S) \;=\; -\sum_{j=1}^{d-1} \underbrace{\frac{N(p_k,\, j+1)}{p_k}}_{[M_d]_{k,j}} e_{j+1}(S), \tag{25}$$

where the second equality uses the index shift $j = q-1$ and the zero convention $[M_d]_{k,j} = 0$ for $j+1 > \min(p_k, d)$. Define the column vectors

$$\mathbf{e} \;\coloneqq\; \big(e_2(S),\, e_3(S),\, \ldots,\, e_d(S)\big)^\top \;\in\; \mathbb{Q}^{d-1}, \qquad \boldsymbol{\delta} \;\coloneqq\; \big(\delta_{p_1}(A_S),\, \ldots,\, \delta_{p_{d-1}}(A_S)\big)^\top \;\in\; \mathbb{Q}^{d-1}.$$

Equation (25), written for each $k = 1, \ldots, d-1$ simultaneously, reads

$$M_d\, \mathbf{e} \;=\; -\boldsymbol{\delta}, \tag{26}$$

which is stated in Definition 1.

(2) We establish that $M_d \in \mathbb{Q}^{(d-1)\times(d-1)}$ is invertible.

Lemma C.2 (Invertibility of M_d)**.** *M_d is invertible over $\mathbb{Q}$.*

Proof of Lemma C.2. Write $M_d = P\,B\,R$, where $P = \mathrm{diag}(1/p_1, \ldots, 1/p_{d-1})$ and $R = \mathrm{diag}(2!, 3!, \ldots, d!)$ are invertible diagonal matrices and

$$[B]_{k,j} \;=\; S(p_k,\, j+1), \qquad k, j \in \{1, \ldots, d-1\}. \tag{27}$$

Thus, $\det(M_d) = \big(\prod_k 1/p_k\big)\big(\prod_j (j+1)!\big)\det(B)$, so M_d is invertible if and only if B is.

Using the inclusion–exclusion formula (21) for Stirling numbers, $S(p,q) = \frac{1}{q!}\sum_{j=0}^{q}(-1)^{q-j}\binom{q}{j}j^p$, we may write $B = V\,C$, where $[V]_{k,j} = j^{p_k}$ for $k \in [d{-}1]$ and $j \in \{1,\dots,d\}$ is a $(d{-}1)\times d$ matrix and C is a $d\times(d{-}1)$ matrix with rational entries determined by binomial coefficients; in particular, $\operatorname{rank}(B) \le \operatorname{rank}(V)$.

We show V has full row rank $d{-}1$. Suppose $\sum_{k=1}^{d-1} c_k\, j^{p_k} = 0$ for all $j \in \{1,\dots,d\}$. Define the function $f : \{1,\dots,d\} \to \mathbb{Q}$, $f(j) = \sum_k c_k j^{p_k}$. Viewing f as determined by the vector $(f(1),\dots,f(d)) \in \mathbb{Q}^d$ and using the basis $\{j^{p_1},\dots,j^{p_{d-1}},\mathbf{1}\}$ (where $\mathbf{1}$ is the constant function), the matrix $[G]_{j,k} = j^{p_k}$ for $j \in \{1,\dots,d\}$, $k \in [d{-}1]$ is a sub-matrix of the $(d\times d)$ generalized Vandermonde matrix $[\tilde{G}]_{j,k} = j^{r_k}$ with exponents $r_0 = 0 < p_1 < p_2 < \cdots < p_{d-1}$ evaluated at the d distinct positive integer nodes $j = 1,\dots,d$.

The determinant of $[\tilde{G}]$ factors as (see Chapter 7.16 in [42])

$$\det\big([\tilde{G}]_{j,k}\big) \;=\; \prod_{1\le i<j\le d}(j-i)\;\cdot\; s_\lambda(1,2,\dots,d) \;\neq\; 0,$$

where s_λ is the Schur polynomial of the partition $\lambda_k = r_{d-k} - (d{-}k)$, which has strictly positive value at positive arguments. Since $\tilde{G}$ is invertible and the sub-matrix G (obtained by removing the column $r_0 = 0$, i.e. the all-ones column) has full column rank $d{-}1$, the system $G\mathbf{c} = \mathbf{0}$ has only the trivial solution. Hence V has full row rank and therefore $\operatorname{rank}(B) = d{-}1$, i.e. B is invertible. Consequently M_d is invertible. □

Remark 4 (Alternative verification for small d)**.** For practical degrees $d \le 5$, the matrices M_d and their determinants may be verified by direct computation. For $d=2$: $M_2 = (1)$, $\det = 1$. For $d=3$: $M_3 = \left(\begin{smallmatrix}1&0\\2&2\end{smallmatrix}\right)$, $\det = 2$. For $d=4$: $M_4 = \left(\begin{smallmatrix}1&0&0\\2&2&0\\6&30&48\end{smallmatrix}\right)$, $\det = 96$. For $d=5$: $\det(M_5) = 172{,}800 \neq 0$. In each case M_d is lower-triangular (for $d \le 4$) or nearly so, and the diagonal entries $[M_d]_{k,k} = N(p_k, k{+}1)/p_k$ are strictly positive.

(3) Since M_d is invertible, Eq. (26) has the unique solution

$$\mathbf{e} \;=\; -M_d^{-1}\,\boldsymbol{\delta}. \tag{28}$$

Component-wise, the ℓ-th entry ($\ell \in [d{-}1]$) reads

$$e_{\ell+1}(S) \;=\; \sum_{k=1}^{d-1}\big[-M_d^{-1}\big]_{\ell,k}\,\delta_{p_k}(A_S). \tag{29}$$

(4) The degree-d monomial is $e_d(S) = \prod_{i\in S} x_i$, which corresponds to $\ell = d{-}1$ in (29):

$$\prod_{i\in S} x_i \;=\; e_d(S) \;=\; \sum_{k=1}^{d-1}\big[-M_d^{-1}\big]_{d-1,\,k}\,\delta_{p_k}(A_S). \tag{30}$$

Defining $\alpha_k^{(d)} \coloneqq [-M_d^{-1}]_{d-1,k}$ (the $(d{-}1)$-th row of $-M_d^{-1}$) yields the representation

$$\prod_{i\in S} x_i \;=\; \sum_{k=1}^{d-1}\alpha_k^{(d)}\,\delta_{p_k}(A_S),$$

exactly as stated. The coefficients $\boldsymbol{\alpha}^{(d)} = (\alpha_1^{(d)},\dots,\alpha_{d-1}^{(d)})^\top$ are rational numbers uniquely determined by $-M_d^{-1}$. Equation (30) holds as an *exact algebraic identity* for every Boolean vector $\mathbf{x} \in \{0,1\}^{|S|}$. □

C.3 Proof of Lemma 3.3

Proof. We first establish the rightmost equality. Since $x_i \in \{0,1\}$, we have $\prod_{i\in T} x_i = 1$ if and only if $x_i = 1$ for all $i \in T$, which holds if and only if $A_T = \sum_{i\in T} x_i = m$. Hence $\prod_{i\in T} x_i = \mathbf{1}[A_T = m]$. It remains to show $q^T = \mathbf{1}[A_T = m]$ in each case.

Case 1. ($m \le 5$, domain-wall chain). The validity constraints (11) force the unique configuration $q_j^T = \mathbf{1}[j \le A_T]$ for each $j = 1, \dots, m$. The output qubit is $q^T := q_m^T = \mathbf{1}[m \le A_T]$. Since $A_T \in \{0, \dots, m\}$, the condition $m \le A_T$ is equivalent to $A_T = m$, so $q^T = \mathbf{1}[A_T = m]$.

Case 2. ($m \ge 6$, binary-counter chain). Let $\ell = \lceil \log_2(m+1) \rceil$ and write the binary representation $m = \sum_{j=0}^{\ell-1} m_j 2^j$. On a valid chain state, $A_T = \sum_{j=0}^{\ell-1} 2^j r_j^T$, so $A_T = m$ if and only if $r_j^T = m_j$ for all j. By definition (13), $c_j^T = 1$ if and only if $r_j^T = m_j$. Therefore $A_T = m \iff c_j^T = 1$ for all $j = 0, \dots, \ell - 1$.

We claim $a_k^T = \prod_{j=0}^{k} c_j^T$ for $k = 1, \dots, \ell-2$ by induction. For the base case, $a_1^T = c_0^T c_1^T = \prod_{j=0}^{1} c_j^T$. Assuming $a_{k-1}^T = \prod_{j=0}^{k-1} c_j^T$, the recurrence gives $a_k^T = a_{k-1}^T \cdot c_k^T = \prod_{j=0}^{k} c_j^T$. Since each product a_k^T is enforced by the penalty $P(u, v, w) = uv - 2uw - 2vw + 3w \ge 0$, which vanishes if and only if $w = uv$, the conjunction is exact. Applying the final step,

$$q^T = a_{\ell-2}^T \cdot c_{\ell-1}^T = \prod_{j=0}^{\ell-1} c_j^T = \mathbf{1}\big[\, \forall j : c_j^T = 1 \,\big] = \mathbf{1}[A_T = m].$$

In both cases $q^T = \mathbf{1}[A_T = m] = \prod_{i \in T} x_i$. □

C.4 Proof of Theorem 3.4

Proof. Let $S = T_S \cup \{v_S\}$ with $T_S \cap \{v_S\} = \emptyset$ and $|T_S| = d - 1$. On every valid joint state of $\mathcal{H}_{T_S}$, Lemma 3.3 gives $q^{T_S} = \prod_{i \in T_S} x_i$. Therefore,

$$x_{v_S} \cdot q^{T_S} = x_{v_S} \cdot \prod_{i \in T_S} x_i = \prod_{i \in S} x_i,$$

where the last equality uses $S = T_S \cup \{v_S\}$ and $x_{v_S}^2 = x_{v_S}$ (idempotency over $\{0, 1\}$). The right-hand side $x_{v_S} \cdot q^{T_S}$ is a product of exactly two binary variables, constituting a single degree-2 QUBO coupler. □

C.5 Proof of Corollary 3.5

Proof. **(i)** Let $\lambda > \sum_S |c_S|$. The QUBO objective is $\widetilde{E}(\mathbf{x}, \mathbf{q}) = E_{\text{quad}}(\mathbf{x}, \mathbf{q}) + \lambda \sum_T V_T(\mathbf{q})$, where $V_T \ge 0$ is the total penalty of chain $\mathcal{H}_T$ and E_{quad} encodes the couplers from Theorem 3.4. For any $\mathbf{x}$, any invalid chain state satisfies $V_T \ge 1$, contributing at least λ to $\widetilde{E}$. Since $\lambda > \sum_S |c_S| \ge |E_{\text{quad}}(\mathbf{x}, \mathbf{q}) - E(\mathbf{x})|$ for all $(\mathbf{x}, \mathbf{q})$, the penalty strictly dominates any possible gain from a chain violation. Hence at any minimizer $(\mathbf{x}^*, \mathbf{q}^*)$ all chains are valid, Theorem 3.4 applies to every monomial, and $\widetilde{E}(\mathbf{x}^*, \mathbf{q}^*) = E(\mathbf{x}^*)$. It follows that $\min_{\mathbf{x}} \widetilde{E}(\mathbf{x}, \mathbf{q}^*) = \min_{\mathbf{x}} E(\mathbf{x})$.

(ii) By Theorem 3.4, each $S \in \mathcal{H}$ is replaced by exactly the single coupler $x_{v_S} q^{T_S}$, regardless of $|S|$. The construction introduces no other degree-2 terms from S itself.

(iii) For a sub-aggregate T with $|T| = m$: **(1)** For $m \le 5$, the domain-wall chain introduces m ancilla qubits $q_1^T, \dots, q_m^T$, so $\nu(m) = m$. **(2)** For $m \ge 6$, let $\ell = \lceil \log_2(m+1) \rceil$. The binary-counter chain uses ℓ register bits $\mathbf{r}^T$, $\ell - 2$ intermediate conjunction ancillas $a_1^T, \dots, a_{\ell-2}^T$, and 1 output qubit q^T, giving $\ell + (\ell - 2) + 1 = 2\ell - 1 = 2\lceil \log_2(m+1) \rceil - 1$ ancilla qubits. The crossover occurs at $m = 5$: $2\lceil \log_2(6) \rceil - 1 = 2(3) - 1 = 5 = m$, confirming continuity at the boundary. For $m \ge 6$, $2\lceil \log_2(m+1) \rceil - 1 < m$. Hence $\nu(m) = \min\big(m,\ 2\lceil \log_2(m+1) \rceil - 1\big)$.

(iv) If $T_{S'} = T$ for some existing chain $\mathcal{H}_T$, then q^T is already constructed and available. Theorem 3.4 yields $\prod_{i \in S'} x_i = x_{v_{S'}} \cdot q^T$, which contributes exactly one new coupler $x_{v_{S'}} q^T$ to $\widetilde{E}$. No new ancilla qubits are required since $\mathcal{H}_T$ is unchanged.

□

D Supplemental materials of Examples

D.1 Computation of δ-values

Given a quartic energy function in Example 1

$$E(\mathbf{x}) = 5x_1x_2x_3x_4 - 3x_1x_2x_3 + x_1x_2 - 2x_3. \tag{31}$$

Let $a_{1234} = (x_1{+}x_2{+}x_3{+}x_4)$, $a_{123} = (x_1{+}x_2{+}x_3)$, $a_{12} = (x_1{+}x_2)$. We define $\delta_p(a) = (a - a^p)/p$, for p is a prime and compute

$$\delta_2(a_{12}) = \frac{(x_1+x_2) - (x_1+x_2)^2}{2} = \frac{-2x_1x_2}{2} = -x_1x_2 \tag{32a}$$

$$\delta_2(a_{123}) = \frac{(x_1+x_2+x_3) - (x_1+x_2+x_3)^2}{2} = -(x_1x_2 + x_1x_3 + x_2x_3) \tag{32b}$$

$$\delta_3(a_{123}) = \frac{(x_1+x_2+x_3) - (x_1+x_2+x_3)^3}{3} = -2(x_1x_2 + x_1x_3 + x_2x_3) - 2x_1x_2x_3 \tag{32c}$$

$$\delta_2(a_{1234}) = -(x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4) \tag{32d}$$

$$\delta_3(a_{1234}) = 2\,\delta_2(a_{1234}) - 2(x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4) \tag{32e}$$

$$\delta_5(a_{1234}) = 6\,\delta_2(a_{1234}) - 30(x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4) - 48\,x_1x_2x_3x_4 \tag{32f}$$

D.2 Binary-counter HOBO-to-QUBO encoding - worked examples

Case A: Degree-5 monomial, domain-wall chain ($m = 4$)]

Consider

$$E(x_1, \ldots, x_5) = 3\,x_1x_2x_3x_4x_5, \qquad x_i \in \{0, 1\}.$$

$E = 0$ everywhere except $E = 3$ at $\mathbf{x} = (1, 1, 1, 1, 1)$. Let $S = \{1, 2, 3, 4, 5\}$, $d = 5$, $w_S = 3$, we choose:

$$T_S = \{1, 2, 3, 4\}, \quad v_S = x_5, \quad m = |T_S| = 4.$$

By (17), $\nu(4) = \min\big(4,\ 2\lceil \log_2 5 \rceil - 1\big) = \min(4, 5) = 4$ ancilla qubits q_1, q_2, q_3, q_4.

A_T	(q_1, q_2, q_3, q_4)	$q^T := q_4$
0	$(0, 0, 0, 0)$	0
1	$(1, 0, 0, 0)$	0
2	$(1, 1, 0, 0)$	0
3	$(1, 1, 1, 0)$	0
4	$(1, 1, 1, 1)$	1

Besides, $q_1 \geq q_2 \geq q_3 \geq q_4$ and $\sum_{j=1}^{4} q_j = A_T := x_1 + x_2 + x_3 + x_4$. Lemma 3.3 is verified by the beside table; i.e., $q_4 = 1 \iff A_T = 4 \iff x_1x_2x_3x_4 = 1$. By Theorem 3.4, $x_1x_2x_3x_4x_5 = x_5 \cdot q_4$. Penalty terms are

$$P_{\text{ord}} = (q_2 - q_1q_2) + (q_3 - q_2q_3) + (q_4 - q_3q_4). \tag{33a}$$

$$P_{\text{sum}} = \big(q_1 + q_2 + q_3 + q_4 - x_1 - x_2 - x_3 - x_4\big)^2. \tag{33b}$$

Expanding P_{sum} with $y^2 = y$:

$$P_{\text{sum}} = (x_1 + x_2 + x_3 + x_4 + q_1 + q_2 + q_3 + q_4) + 2(x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4)$$
$$+ 2(q_1q_2 + q_1q_3 + q_1q_4 + q_2q_3 + q_2q_4 + q_3q_4) - 2 \sum_{\substack{i \in \{1,2,3,4\} \\ j \in \{1,2,3,4\}}} q_j x_i.$$

The reformulated energy function is $\tilde{E}(\mathbf{x}, \mathbf{q}) = 3\,x_5q_4 + 4\,P_{\text{ord}} + 4\,P_{\text{sum}}$. and the QUBO matrix is given as

$$Q = \begin{pmatrix} 4 & 8 & 8 & 8 & 0 & -8 & -8 & -8 & -8 \\ & 4 & 8 & 8 & 0 & -8 & -8 & -8 & -8 \\ & & 4 & 8 & 0 & -8 & -8 & -8 & -8 \\ & & & 4 & 0 & -8 & -8 & -8 & -8 \\ & & & & 0 & 0 & 0 & 0 & 3 \\ & & & & & 4 & 4 & 8 & 8 \\ & & & & & & 8 & 4 & 8 \\ & & & & & & & 8 & 4 \\ & & & & & & & & 8 \end{pmatrix}$$

To verify, we compute

$(x_1, \ldots, x_5)$	(q_1, q_2, q_3, q_4)	$3x_5q_4$	$4P_{\text{ord}}$	$4P_{\text{sum}}$	$\tilde{E}$
$(0,0,0,0,0)$	$(0,0,0,0)$	0	0	0	**0**
$(1,0,0,0,0)$	$(1,0,0,0)$	0	0	0	**0**
$(1,1,0,0,1)$	$(1,1,0,0)$	0	0	0	**0**
$(1,1,1,0,1)$	$(1,1,1,0)$	0	0	0	**0**
$(1,1,1,1,0)$	$(1,1,1,1)$	0	0	0	**0**
$(1,1,1,1,1)$	$(1,1,1,1)$	3	0	0	**3**

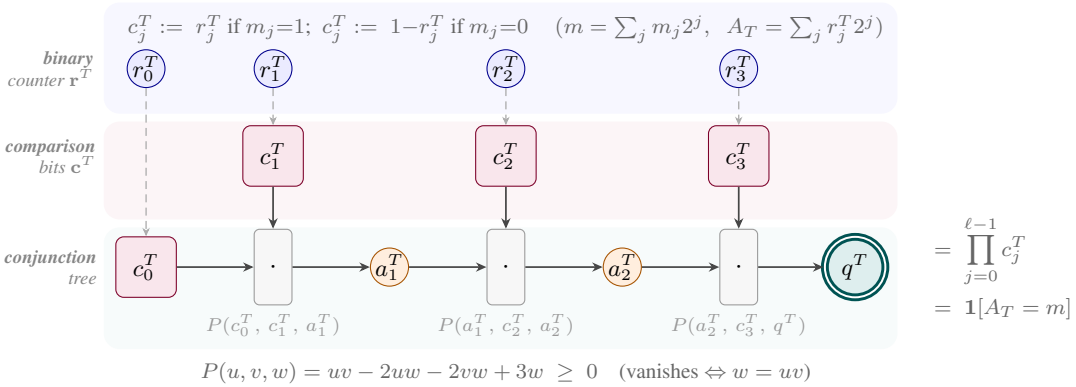
At $\mathbf{x} = (1,1,1,1,1)$ with the qubit chain $= (0,0,0,1)$, where violating ordering and sum:

$$4P_{\text{ord}} = 4 \cdot [0 + 0 + 1(1-0)] = 4, \quad 4P_{\text{sum}} = 4(1-4)^2 = 36, \quad \tilde{E} = 0 + 4 + 36 = 40 \gg 3.$$

Degree-9 monomial, binary-counter chain ($m = 8$, $\ell = 4$) Consider

$$E(x_1, \ldots, x_9) = x_1x_2x_3x_4x_5x_6x_7x_8x_9, \qquad x_i \in \{0,1\}.$$

Let $T_S = \{1, \ldots, 8\}$, $v_S = x_9$, $m = 8$. Since $m = 8 > 5$, we use the binary-counter chain. The counter bit-width is $\ell = \lceil \log_2(m+1) \rceil = \lceil \log_2 9 \rceil = 4$. We present the encoding procedures in below diagram, with corresponding details highlight by the same color.



Binary ancillas $r_0, r_1, r_2, r_3 \in \{0,1\}$ encode:

$$A_T = \sum_{i=1}^{8} x_i = r_0 \cdot 2^0 + r_1 \cdot 2^1 + r_2 \cdot 2^2 + r_3 \cdot 2^3.$$

The target sum is $A_T = 8 = (1000)_2$, so $m_0 = 0,\ m_1 = 0,\ m_2 = 0,\ m_3 = 1$. The **comparison bits** are $c_0 = 1 - r_0, \quad c_1 = 1 - r_1, \quad c_2 = 1 - r_2, \quad c_3 = r_3$. Here, $c_j = 1 \iff$ bit j of A_T equals bit j of $m = 8$. The **conjunction tree** is constructed by $\ell - 2 = 2$ intermediate ancillas as

$$a_1 = c_0 \cdot c_1 = (1 - r_0)(1 - r_1), \quad a_2 = a_1 \cdot c_2 = a_1(1 - r_2), \quad q^T = a_2 \cdot c_3 = a_2\, r_3.$$

We verify three values $A_T \in \{7, 8, 9\}$ as

$$A_T = 7 = (0111)_2 : \ (r_0, r_1, r_2, r_3) = (1,1,1,0) \ \Rightarrow \ c_0 = 0 \ \Rightarrow \ a_1 = 0,\ q^T = 0. =$$

$$A_T = 8 : \ (r_0, r_1, r_2, r_3) = (0,0,0,1) \ \Rightarrow \ c_0 = c_1 = c_2 = c_3 = 1 \ \Rightarrow \ a_1 = a_2 = q^T = 1.$$

$$A_T = 9 \text{ (impossible for 8 bits)} : \text{constraint enforced by } P_{\text{bc}}.$$

For ancilla count, we use 4 counter qubits, 2 intermediate conjunction and 1 output qubit; in total of 7 ancillas. This is consistent to (17) as $\nu(8) = \min(8,\ 2 \cdot 4 - 1) = 7$. Next, we construct penalty terms, which consists of two groups:

Binary counter penalties enforces $A_T = \sum_j 2^j r_j$, we have

$$P_{\text{bc}} = \left(\textstyle\sum_{i=1}^{8} x_i - r_0 - 2r_1 - 4r_2 - 8r_3 \right)^2.$$

After expansion with $y^2 = y$:

$$P_{\text{bc}} = \sum_{i=1}^{8} x_i + r_0 + 4r_1 + 16r_2 + 64r_3 + 2\sum_{1\le i<j\le 8} x_i x_j + 2(r_0 r_1 \cdot 2 + r_0 r_2 \cdot 4 + r_0 r_3 \cdot 8 + r_1 r_2 \cdot 8 + r_1 r_3 \cdot 16 + r_2 r_3 \cdot 32) - 2\Big(\sum_{i=1}^{8} x_i\Big)\big(r_0 + 2r_1 + 4r_2 + 8r_3\big).$$

Conjunction penalties for a_1, a_2 and q are

$$P_{a_1} = -r_0 - r_1 + r_0 r_1 - a_1 + 2r_0 a_1 + 2r_1 a_1$$
$$P_{a_2} = a_1(1 - r_2) - 2a_1 a_2 - 2(1 - r_2)a_2 + 3a_2$$
$$P_{q^T} = a_2 r_3 - 2a_2 q^T - 2r_3 q^T + 3q^T.$$

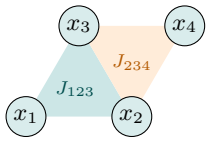
The final QUBO energy function is $\tilde{E}(\mathbf{x}, \mathbf{q}) = x_9\, q^T + 2\big(P_{\text{bc}} + P_{a_1} + P_{a_2} + P_{q^T}\big)$.

We emphasise that FerQ-Bc's qubit-overhead advantage is conditional on the problem structure; however, when the design conditions are satisfied, this advantage is guaranteed by construction. Specifically, when multiple high-degree monomials share a common sub-aggregate, e.g., $E(x) = c_1 x_1 x_2 x_3 x_4 + c_2 x_1 x_2 x_3 x_5 + c_3 x_1 x_2 x_3 x_6$, FerQ-Bc introduces a *single auxiliary variable* for the shared product $x_1 x_2 x_3$ and reuses it across all monomials, whereas Rosenberg quadratises each monomial independently.

E Supplemental materials of Results

E.1 Random p-spin glass model

The p-spin glass model minimizes $E(\mathbf{x}) = \sum_{\substack{S\subseteq[n]\\|S|=p}} J_S \prod_{i\in S} x_i$ over binary variables $x \in \{0,1\}^n$, where the interactions J_S are Gaussian-distributed. Since $p > 2$ renders direct quadratic optimisation inapplicable, the goal is to evaluate how faithfully different reformulation methods recover the true ground-state energy after reducing to 2-local (quadratic) form.

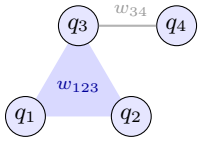


$E(\boldsymbol{x}) = J_{123}\, x_1 x_2 x_3 + J_{234}\, x_2 x_3 x_4$

Random p-spin instances in Section 4.2.2 with Gaussian interaction strengths are generated over $N \in \{6, 8, 10, 12, 14, 16\}$ variables and interaction orders $p \in \{3, 5, 7, 9, 11\}$, subject to $p \le N$, yielding $\min(\max(5, 2N), \binom{N}{p})$ interactions per instance. Ten independent instances are generated per (N, p) configuration. The ground-truth energy E^* is established via brute-force enumeration over all 2^N binary assignments. Each reformulation method is applied once per instance and evaluated on all three metrics as N scales.

E.2 k-local Hamiltonian

A k-local Hamiltonian ($k > 2$) must be reduced to 2-local form to enable practical optimisation. The beside illustrates a k-local Hamiltonian ($k = 3$, $N = 4$). The shaded triangle denotes a 3-body interaction term. The optimizer seeks a binary assignment $q_i \in \{|0\rangle, |1\rangle\}$ that minimizes H. The goal is to evaluate how faithfully different reformulation methods preserve the ground-state energy after this transformation. We evaluate methods using: (1) average spectral error $|E_{\text{truth}} - \hat{E}_0|$, (2) *is-optimal* rate (fraction of exact recoveries), and (3) ancilla variable count.



$H(\boldsymbol{q}) = w_{123}\, q_1 q_2 q_3 + w_{34}\, q_3 q_4$

We generate random k-local Hamiltonians in Section 4.2.3 with Gaussian interaction strengths for $N \in \{6, \ldots, 16\}$ and $k \in \{3, 5, 7, 9, 11\}$. For each (N, k), we sample 10 instances. Ground truth

E_0 is obtained via brute-force enumeration over all 2^N assignments. Each reformulation method is evaluated once per instance across all metrics as N scales.

E.3 GPU acceleration performance of FerQ

We benchmark FerQ implemented on CPU (`NumbaJIT`) and GPU (`CUDA`) backends on synthetic random 3-SAT problems. Problem instances were generated at a clause-to-variable density of 4.0 (near the 3-SAT phase transition) for problem sizes ranging from 50 to $1,000$ variables, and solved an inverse-temperature schedule spanning $\beta \in [0.1, 10]$. Each configuration combined one of five variable counts $(50, 100, 200, 500, 1000)$ with one of four parallel-read counts $(50, 256, 1024, 2048)$, yielding 20 configurations in total, each repeated over three independent trials to obtain mean and minimum wall-clock times. Both backends were warmed up prior to timing to exclude one-time JIT compilation costs, and GPU speedup was computed as the ratio of mean CPU to mean GPU solve time.

F Supplemental Tables

Some surjection numbers $N(p, q)$ and coefficient α_k are shown in Table F.1 and F.2, respectively. Table F.3 shows an example of HOBO reduction using FerQ and compared reducers. The performance comparison between FerQ v.s. un-reformulated problem is reported in Table F.4, and FerQ-Bc v.s. Rosenberg encoding is in Table F.5, with the required number of qubits are reported in Table F.6. Floating-point error and scalability analysis for FerQ's computation are shown in Table F.7 and F.8.

Table F.1: Surjection numbers $N(p, q) = q!\, S(p, q)$ for selected primes p and degrees q.

$p \setminus q$	$q=1$	$q=2$	$q=3$	$q=4$	$q=5$	$q=6$	$q=7$	$q=8$	$q=9$
2	1	2	0	0	0	0	0	0	0
3	1	6	6	0	0	0	0	0	0
5	1	30	150	240	120	0	0	0	0
7	1	126	1806	8400	16800	15120	5040	0	0
11	1	2046	171006	3498000	29607600	129230640	322494480	479001600	419126400
13	1	8190	1569750	60780720	901020120	6711344640	28805736960	76592355840	130456085760
17	1	131070	128746950	16664094960	678330198120	12604139926560	129568848121440	823172919528960	3457819037312640

Table F.2: Coefficients $\alpha_k^{(d)}$ for degrees $d = 2, \ldots, 6$ using the canonical prime sequence. All values are exact rationals computed by inversion of M_d.

d	Primes $\mathcal{P}_d$	$\alpha^{(d)} = (\alpha_1^{(d)}, \ldots, \alpha_{d-1}^{(d)})$
2	$\{2\}$	(-1)
3	$\{2, 3\}$	$\left(1,\ -\frac{1}{2}\right)$
4	$\{2, 3, 5\}$	$\left(-\frac{1}{2},\ \frac{5}{16},\ -\frac{1}{48}\right)$
5	$\{2, 3, 5, 7\}$	$\left(\frac{1}{5},\ -\frac{41}{300},\ \frac{1}{72},\ -\frac{1}{1800}\right)$
6	$\{2, 3, 5, 7, 11\}$	$\left(-\frac{1}{24},\ \frac{101}{3456},\ -\frac{2855}{870912},\ \frac{67}{414720},\ -\frac{1}{8709120}\right)$

Table F.3: We generate 100 random HOBO energy functions. The number of high-ordered monomial is 16 (**Input HO**). **Residual HO** is the mean number of degree-≥ 3 terms left in the output. All energy transformation methods use the same solver configuration. All baseline reductions (except Rosenberg) only partially reduce the degree of high-order monomials, leaving a higher-order objective. Only FerQ-Bc and Rosenberg produce quadratic energy functions.

Method	**Input HO**	**Residual HO**	**Mean Out-degree**
FerQ-Bc	16.0	0.00 ± 0.00	2.00
Rosenberg	16.0	0.00 ± 0.00	2.00
DeducReduc	16.0	10.80 ± 2.89	14.57
NTR_KZFD	16.0	8.33 ± 1.93	14.56
NTR_ABCG	16.0	8.33 ± 1.93	14.56
NTR_ABCG2	16.0	8.33 ± 1.93	14.56
NTR_GBP	16.0	15.39 ± 0.75	15.52
PTR_Ishikawa	16.0	7.67 ± 1.93	14.78
PTR_KZ	16.0	15.36 ± 0.77	15.52
PTR_GBP	16.0	15.36 ± 0.77	15.52
FGBZ_Negative	16.0	16.93 ± 0.26	15.20
FGBZ_Positive	16.0	21.78 ± 1.47	15.22
PairwiseCovers	16.0	21.08 ± 1.61	14.73
FerQ	16.0	16.00 ± 0.00	15.52

Table F.4: FerQ vs. the un-reformulated raw HOBO.

Application	n	FerQ	Raw HOBO	Wilcoxon p
SAT	54	0.989 ± 0.024	0.990 ± 0.024	0.006**
MaxSAT	56	0.997 ± 0.005	0.997 ± 0.005	0.953
p-spin	240	0.000 ± 0.000	0.000 ± 0.000	0.317
QC-Hamiltonian	240	0.000 ± 0.000	0.000 ± 0.000	1.000

Table F.5: Performance comparison between FerQ-Bc and Rosenberg encodings as QUBO transformations across applications.

Application	Metric	n	FerQ-Bc	Rosenberg	Wilcoxon p
SAT	Sat. rate ↑	54	$\mathbf{0.877 \pm 0.077}$	0.875 ± 0.130	0.163
MaxSAT	Sat. rate ↑	56	$\mathbf{0.964 \pm 0.061}$	0.958 ± 0.093	0.266
p-spin	Rel. error ↓	240	0.613 ± 0.461	$\mathbf{0.339 \pm 0.417}$	8.2e-20***
QC-Hamiltonian	Rel. error ↓	240	$\mathbf{0.656 \pm 0.427}$	1.553 ± 2.076	4.0e-08***

Table F.6: Auxiliary qubits added per application (mean±std) between FerQ-Bc and Rosenberg reduction.

Method	SAT	MaxSAT	p-spin	QC-Hamiltonian
FerQ-Bc	10505±28802	3414±14595	81±55	82±56
Rosenberg	15843±46408	7202±31274	99±78	99±78

Table F.7: Numerical analysis of floating-point error.

d	$\mathrm{err}(\mathrm{exact} \to \mathrm{f64})$	$\kappa(M_d)$
2	0.000×10^{0}	1.00×10^{0}
3	0.000×10^{0}	4.27×10^{0}
4	0.000×10^{0}	8.92×10^{1}
5	8.882×10^{-16}	4.39×10^{3}
6	2.220×10^{-16}	1.99×10^{7}
7	2.776×10^{-17}	3.80×10^{9}
8	2.220×10^{-16}	8.22×10^{13}
9	3.553×10^{-15}	3.50×10^{16}
10	1.776×10^{-15}	8.01×10^{20}
11	2.665×10^{-15}	4.20×10^{27}
12	5.551×10^{-17}	2.00×10^{30}
13	2.082×10^{-17}	1.12×10^{39}
14	2.220×10^{-16}	6.82×10^{44}
15	5.551×10^{-17}	2.88×10^{45}
18	7.816×10^{-14}	2.23×10^{70}
21	3.695×10^{-13}	3.73×10^{86}
24	1.080×10^{-12}	1.01×10^{105}
27	1.137×10^{-13}	4.45×10^{131}
30	7.276×10^{-12}	1.01×10^{146}
33	3.638×10^{-12}	8.75×10^{179}
36	1.023×10^{-12}	2.32×10^{212}
39	4.275×10^{-11}	9.14×10^{234}
42	8.004×10^{-11}	4.13×10^{263}
43	2.619×10^{-10}	6.71×10^{266}

Table F.8: **Notation.** n: number of *original* binary variables of the HOBO $E(\mathbf{x})$; m: number of terms with nonzero coefficient, $m = |\{S : c_S \neq 0\}|$; c_S: coefficient of the monomial $\prod_{i \in S} x_i$ in $E(\mathbf{x})$; $\sum |c_S|$: total coefficient mass; λ: global penalty weight in $\tilde{E}$; ρ: relative penalty, $\rho = \lambda / \sum |c_S|$ (how large λ is vs. objective mass); $\rho_{\min}$: smallest ρ; D: **dynamic range**, $D = \lambda / \min |c_S|$; bits: coupler precision needed, $\lceil \log_2 D \rceil$. This table shows that: (1) bound sufficient & near-tight ($\rho = 1$ suffices); i.e., $\rho \leq 1$ everywhere (max $\rho_{\min} = 1.000$); (2) $\rho = \lambda / \sum |c|$ is $O(1) \to 1$; (3) only $D = \lambda / \min |c|$ grows, $O(m)$: $D/m \approx 5.01$ (const) $\Rightarrow$ D linear in m.

n	m	$\sum \lvert c_S \rvert$	$\min \lvert c_S \rvert$	ρ	D	bits
50	91	426	1	1.002347	427.0	9
100	474	2402	1	1.000416	2403.0	12
200	1905	9469	1	1.000106	9470.0	14
500	9509	47729	1	1.000021	47730.0	16
1000	47415	237317	1	1.000004	237318.0	18
1000	189494	948788	1	1.000001	948789.0	20