Supplementary Information

Exact Spin Elimination for Quadratic and k-Local Ising Optimization

Natalia G. Berloff

Supplementary Note 1 Relation to prior variable-reduction methods

Variable-reduction methods differ in how they choose spins to remove and in what they preserve. Supplementary Table 1 compares these choices and guarantees. Here, SPVAR means sample-persistence variable reduction, and RG means renormalization group.

Supplementary Table 1: **What variable-reduction methods preserve and require.** The columns distinguish the basis for each reduction, the variables it can remove, and its main limitation. "Exact" means that the reduction preserves the global minimum under the stated assumptions and in exact arithmetic; it does not imply faster search. Roof-duality checks used in this study are described in Supplementary Note 3.6.

Method	Basis for reduction	Exact	Variables removed	Main limitation
Roof duality	Certificate from a relaxation	yes	Spins whose values are persistent	It can fix no spins when the required certificate is absent; the implementation used here is validated in Supplementary Note 3.6 [1, 2].
Constraint-based spin reduction	Exact linear equality constraints	yes	Variables determined by an independent set	It requires explicit equality constraints [3].
SPVAR and correlation fixing	Low-energy samples or measured correlations	no	Spins that appear stable in the sample set	A wrong fixing decision can remove the true optimum [4, 5, 6].
RG decimation	Dominant local fields or couplings	usually no	Selected spins or clusters	The effective Hamiltonian is generally approximate [7, 8].
Pseudo-Boolean basic algorithm	Conditional minimization and back-substitution	yes	Any selected variable	Intermediate polynomials can grow exponentially [9, 1].
Min-sum bucket elimination	Exact minimization over a variable	yes	Any selected variable	Intermediate factors can grow exponentially with induced width; stopping early leaves an exact residual [10].
Sparse spin-glass reductions	Exact local spin reduction	yes	Low-degree spins in the pairwise rules; arbitrary spins in the general construction	General elimination has exponential neighbourhood cost; the tested low-degree rules retain pairwise form [11, 12, 13].
Walsh elimination	Exact local minimization followed by a Walsh transform	yes	Any selected spin	Transform cost and term count can grow exponentially with the neighbour count; induced interaction order is at most that count.

The exact elimination methods minimize over removed spins for each fixed configuration of the remaining spins, as reviewed by Boros and Hammer [1]. The tested sparse spin-glass rules retain pairwise interactions, but their general expansion also permits higher-order terms [12]. Here, we retain those terms within stated resource limits and pass the reduced polynomial to solvers that accept higher-order interactions directly. The performance comparison includes preprocessing. The cubic-graph theorem separately bounds the spin count and couplings of a pairwise reduction.

Supplementary Note 2 Worked Walsh transform and weighted low-degree rules

2.1 Worked transform

We apply main-text Eq. (10) to a spin coupled equally to three neighbours. The calculation below gives every table entry and Walsh coefficient. Supplementary Note 2.2 then gives the rules for unequal couplings.

Consider the function

$$F(s_1, s_2, s_3) = -\left|s_1 + s_2 + s_3\right|. \tag{S1}$$

Following the convention used in the code, define $b_j = (1 + s_j)/2$ and index configurations by $x = b_1 + 2b_2 + 4b_3$. The bit associated with s_1 is therefore the least significant bit, with zero representing spin -1 and one representing $+1$. The value vector is

$$\mathbf{f} = (-3, -1, -1, -1, -1, -1, -1, -3), \tag{S2}$$

and the unnormalized spin-basis Walsh transform is

$$\mathcal{W}_3\mathbf{f} = (-12, 0, 0, -4, 0, -4, -4, 0). \tag{S3}$$

The output index u identifies the spin product: it contains spin j exactly when bit $j-1$ of u is one. Write this subset as $U(u)$. The outputs therefore occur in the order $\varnothing$, $\{1\}$, $\{2\}$, $\{1,2\}$, $\{3\}$, $\{1,3\}$, $\{2,3\}$, $\{1,2,3\}$, with $(\mathcal{W}_3)_{u,x} = \prod_{j\in U(u)} s_j(x)$.

A binary fast Walsh–Hadamard transform (FWHT) uses $(-1)^{\sum_j u_j b_j}$, where u_j is bit $j-1$ of u. To obtain coefficients in our spin convention, multiply that output by $(-1)^{|U(u)|}$. Changing the bit order also requires relabelling both the spin configurations and the coefficient subsets. Finally, dividing by 2^3 gives

$$-\left|s_1 + s_2 + s_3\right| = -\frac{3}{2} - \frac{1}{2}\left(s_1s_2 + s_1s_3 + s_2s_3\right). \tag{S4}$$

The odd-order outputs vanish, as main-text Corollary 2.2 requires, so the sign conversion leaves this example unchanged. To check the convention for an odd coefficient, consider $F(s_1) = -|2 + s_1|$: its values are $(-1, -3)$ in code order and its coefficients are $(-2, -1)$, giving $F(s_1) = -2 - s_1$.

2.2 Weighted low-degree rules

First consider a spin with local field h_a and couplings J_1 and J_2 to two neighbours. For each neighbour configuration, define

$$f_{\sigma_1\sigma_2} = -\left|h_a + J_1\sigma_1 + J_2\sigma_2\right|, \qquad \sigma_1, \sigma_2 \in \{-1, +1\}. \tag{S5}$$

The exact reduced interaction is

$$F(s_1, s_2) = A + h_1s_1 + h_2s_2 + J_{12}s_1s_2, \tag{S6}$$

with

$$\begin{aligned} A &= \frac{1}{4}\sum_{\sigma_1,\sigma_2} f_{\sigma_1\sigma_2}, & h_1 &= \frac{1}{4}\sum_{\sigma_1,\sigma_2} \sigma_1 f_{\sigma_1\sigma_2}, \\ h_2 &= \frac{1}{4}\sum_{\sigma_1,\sigma_2} \sigma_2 f_{\sigma_1\sigma_2}, & J_{12} &= \frac{1}{4}\sum_{\sigma_1,\sigma_2} \sigma_1\sigma_2 f_{\sigma_1\sigma_2}. \end{aligned} \tag{S7}$$

When $h_a = 0$, the linear terms vanish by main-text Corollary 2.2, and the sums give the closed form

$$A = -\tfrac{1}{2}\left(|J_1 + J_2| + |J_1 - J_2|\right), \qquad J_{12} = -\tfrac{1}{2}\left(|J_1 + J_2| - |J_1 - J_2|\right). \tag{S8}$$

Next consider a spin with couplings J_1, J_2, and J_3 to three neighbours and no local field. Let

$$f_{\boldsymbol{\sigma}} = -\left|J_1\sigma_1 + J_2\sigma_2 + J_3\sigma_3\right|. \tag{S9}$$

Because $f_{-\boldsymbol{\sigma}} = f_{\boldsymbol{\sigma}}$, only even-order terms remain, and the exact expansion is

$$F(s_1, s_2, s_3) = A + J_{12}s_1s_2 + J_{13}s_1s_3 + J_{23}s_2s_3, \tag{S10}$$

where

$$A = \frac{1}{8}\sum_{\boldsymbol{\sigma}\in\{-1,+1\}^3} f_{\boldsymbol{\sigma}}, \qquad J_{ij} = \frac{1}{8}\sum_{\boldsymbol{\sigma}\in\{-1,+1\}^3} \sigma_i\sigma_j f_{\boldsymbol{\sigma}}. \tag{S11}$$

This is the weighted star–triangle rule. A local field breaks the symmetry under reversing all neighbour spins and can introduce local fields and three-spin terms. The general formula in Eq. (10) includes these terms without change.

Supplementary Note 3 Quadratic benchmark: supplementary checks

Unless stated otherwise, these checks use the reference energies $E_{\mathrm{ref}}^{\mathrm{main}}$ described in Supplementary Note 3.8. They include any improvements found by additional reference searches, reported in main-text Section 2.3.3.

3.1 Large-instance energy gaps

At $N = 1000$, no tested solver configuration reaches the reference energy within 5 s. However, the energy gaps decrease when the target fraction of spins to remove increases from $f = 0.2$ to $f = 0.4$ (Supplementary Table 2).

3.2 Time budgets, sample rates, and preprocessing

Across the budgets in Supplementary Table 3, the larger elimination target gives the highest observed success fractions in the harder settings. Where success is already close to one, there is little room for improvement.

Both the number of samples returned per second and preprocessing time affect the comparison (Supplementary Table 4). At $\Delta = 3$, the reduced formulation returns more samples per second. At $\Delta = 5$, its higher-order terms make sampling slower, but more attempts can still succeed within the same total budget. Preprocessing takes a small share of the longer budgets, but up to 54% of the shortest budget at $N = 1000$ (main-text Section 2.3.3).

Allowing more neighbours during elimination can make the new interactions costly to evaluate. At $\Delta = 3$ with $d_{\mathrm{cap}} = 4$, the reduced formulation returns about 1.5 times as many samples per second as the original. At $\Delta = 5$ with $d_{\mathrm{cap}} = 6$, the ratio falls to 0.55–0.75 (Supplementary Table 4). Raising the degree cap in the stored graph to eight creates eight-spin interactions and lowers the ratio to 0.24–0.31; at $\Delta = 8$ it falls to 0.09–0.12. The cost of the new interactions can therefore exceed the saving from fewer spins.

Supplementary Table 2: **Smaller energy gaps with a larger elimination target.** Median gap $\delta E_{\text{main}} = E_{\text{best}} - E_{\text{ref}}^{\text{main}}$ at a total budget of $T = 5$ s, including preprocessing for the reduced formulation. The columns compare the original formulation with target elimination fractions $f = 0.2$ and $f = 0.4$. The reference is certified at $\Delta = 3$, $N = 200$ and is best-known in the other rows.

N	Δ	Original	$f = 0.2$	$f = 0.4$
200	3	0.081	0.000	0.000
500	3	8.217	4.000	0.859
1000	3	26.530	14.888	5.233
200	5	0.000	0.000	0.000
500	5	3.265	0.775	0.400
1000	5	13.069	7.905	3.432

Supplementary Table 3: **Observed success at different time budgets.** Entries give the fraction of successful attempts averaged across instances. Success means reaching the reference within the stated energy tolerance: a certified minimum at $\Delta = 3$, $N \leq 200$, and a best-known energy otherwise. Each reduced attempt includes preprocessing and uses $d_{\text{cap}} = \Delta+1$ with `min_degree` ordering; f is the target fraction of spins to remove.

N	Δ	T (s)	Original	$f = 0.2$	$f = 0.4$
200	3	1	0.095	0.375	0.950
200	3	5	0.360	0.730	1.000
500	3	1	0.000	0.000	0.000
500	3	5	0.000	0.000	0.020
200	5	1	0.835	0.955	0.955
200	5	5	1.000	1.000	1.000
500	5	1	0.020	0.010	0.100
500	5	5	0.000	0.140	0.320

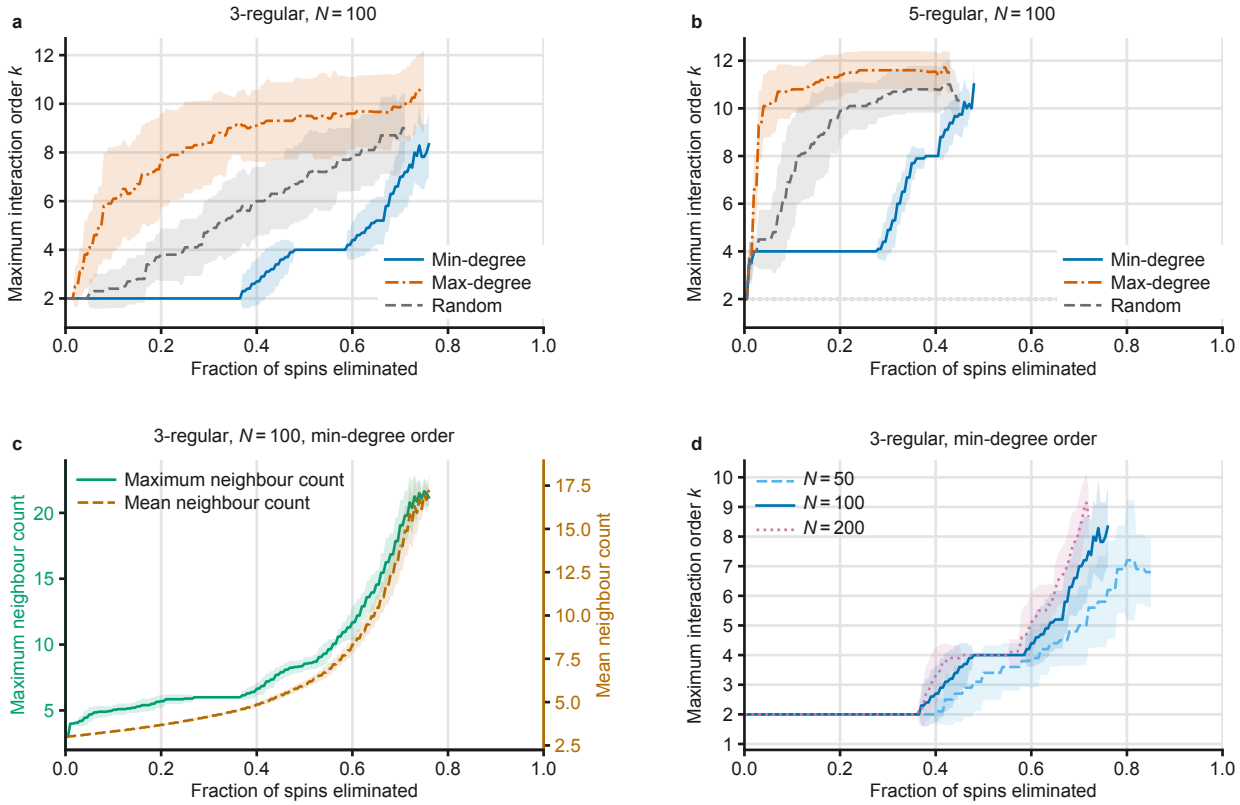
3.3 Growth of interaction order under a neighbour limit

On random 3-regular graphs under `min_degree` ordering, the Hamiltonian stays pairwise until about 37% of the spins are removed (Supplementary Fig. 1): every eliminated spin has at most three neighbours, so main-text Corollary 2.2 applies. The order then rises to four and, beyond about 60%, grows quickly as the surviving spins accumulate neighbours (Supplementary Fig. 1c). The end of the pairwise plateau changes little over the tested sizes (Supplementary Fig. 1d). On 5-regular graphs, the first elimination already creates four-body terms, and the order reaches eight near 35% (Supplementary Fig. 1b). Orderings that remove high-degree spins first raise the order at once.

The default elimination schedule targets $f = 0.4$ with $d_{\text{cap}} = \Delta + 1$ (main-text Section 2.2). At $\Delta = 3$, it stops while the model contains only pairwise or four-body interactions. At $\Delta = 5$, the neighbour limit stops elimination before the target is reached: the removed fraction is 0.30 (Supplementary Fig. 2a), close to the end of the first plateau in Supplementary Fig. 1b.

3.4 Parallel-tempering check

Elimination also improves results with parallel tempering. The test uses 16 copies of the system at temperatures spaced in a geometric sequence, with the same temperature set for both formulations



Supplementary Figure 1: **More elimination can create higher-order interactions. a,b**, Largest interaction order k against the fraction of spins removed from zero-field Gaussian spin glasses on random 3-regular (**a**) and 5-regular (**b**) graphs with $N = 100$. The orders are `min_degree`, `max_degree`, and random, with no limit on the number of terms containing the selected spin. The `max_degree` order removes the highest-degree eligible spin first; it tests an unfavourable choice and is not used in the performance benchmarks. A spin is eligible only when it has at most $d_{\mathrm{cap}} = 12$ neighbours, which limits its Walsh table to 2^{12} entries. This limit applies to the selected spin, not to every remaining spin; a run stops when no eligible spin remains. Lines show means over the runs that reach each removed fraction, starting from 20 instances; bands show one standard deviation across those runs. **c**, Largest (left axis, green) and mean (right axis, orange) neighbour counts among the remaining spins for 3-regular graphs under `min_degree` ordering. **d**, Largest interaction order under `min_degree` ordering for $N = 50$, 100, and 200. Curves are shown only while at least 3 of the 20 runs contribute. For $N = 50$ in **d**, all 20 runs eliminate every spin; its curve is instead truncated at $f = 0.85$, after which the order falls to zero as the last spins are removed. The instances use base seed 42 and the deterministic scheme in `plot_klocality_tradeoff_v38.py`. Values, contributing-run counts, and instance seeds are stored in `klocality_tradeoff_values.npz` and the equivalent `klocality_tradeoff_values.csv`.

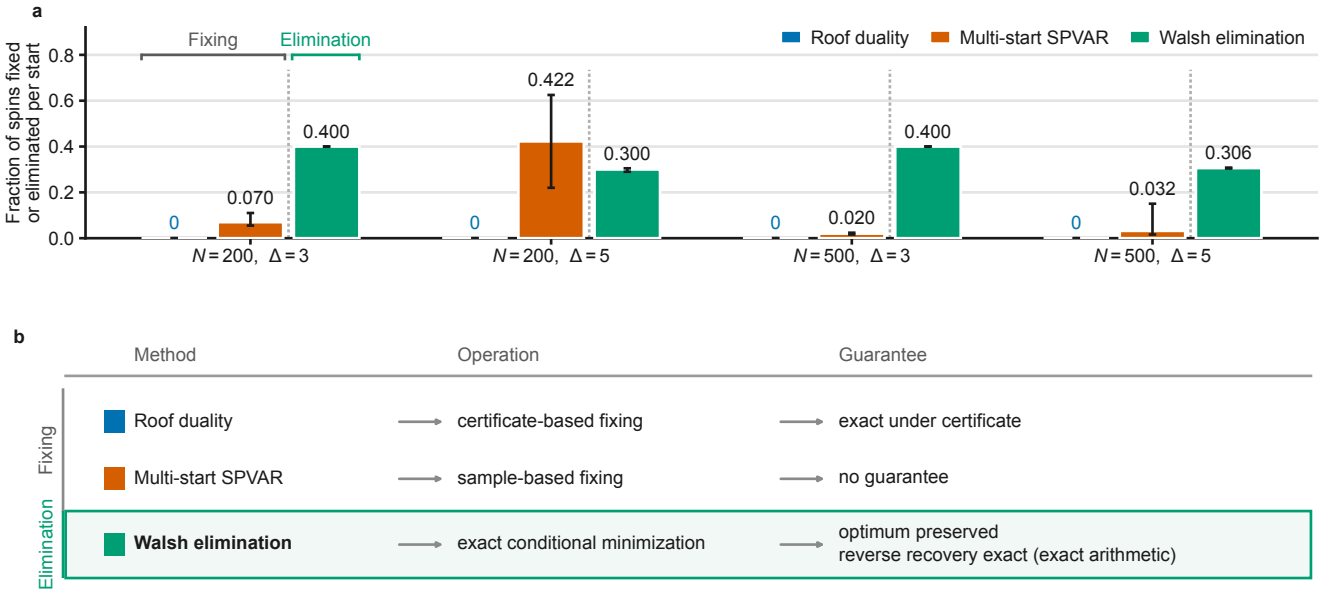
Supplementary Table 4: **Sample rates and preprocessing time.** Results use $T = 5$ s. RPS counts returned samples (reads) per second for the original formulation. Each ratio divides the reduced sample rate by the original rate; values below one mean slower sampling after elimination. The final column gives median preprocessing time for the target fraction $f = 0.4$.

N	Δ	Original RPS	$f = 0.2$ ratio	$f = 0.4$ ratio	t_{prep} (s)
200	3	470	1.21	1.54	0.010
500	3	183	1.16	1.49	0.040
1000	3	109	1.18	1.52	0.140
200	5	445	0.82	0.75	0.020
500	5	181	0.75	0.66	0.085
1000	5	91	0.71	0.55	0.270

(Supplementary Table 5).

3.5 Comparison with fixing methods on new instances

This comparison asks whether preprocessing improves the best energy found on each instance, without requiring a known minimum. Supplementary Note 3.9 gives the protocol, and Supplementary Note 3.6 describes the implementation, reference searches, and checks.



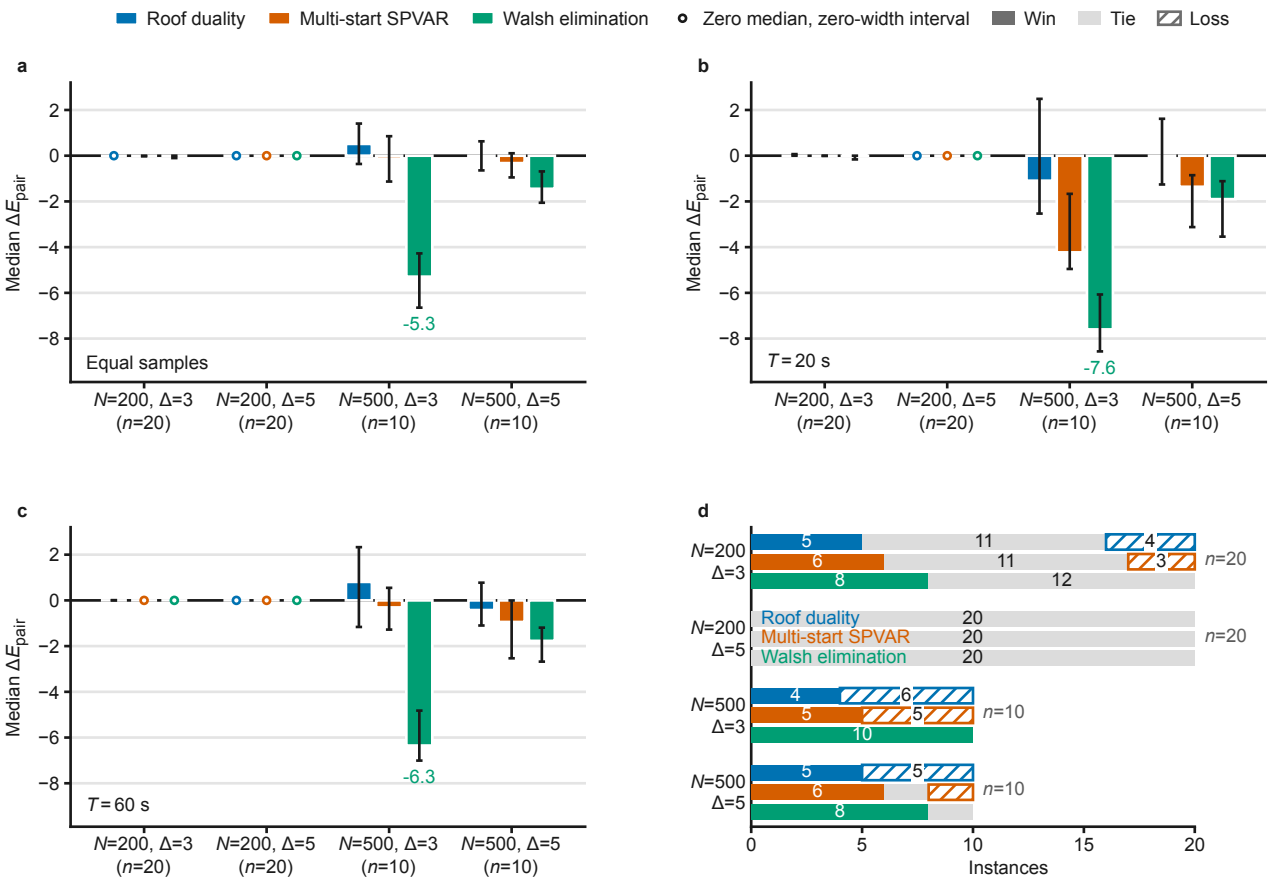
Supplementary Figure 2: **Spin reduction with and without an exact recovery guarantee.** **a**, Fraction of spins fixed or eliminated in each start under the equal-sample protocol. Each instance has 20 starts. The 20 instances at each $N = 200$ setting and 10 at each $N = 500$ setting therefore give 400 and 200 instance–start pairs, respectively. Bars and labels show the median fraction across these pairs; error bars span the middle half of the results (the interquartile range). Roof duality fixes no spins. For SPVAR, the medians are 0.070, 0.422, 0.020, and 0.032 in the displayed setting order. Walsh elimination removes 0.300–0.400 of the spins under the stated schedules. **b**, Roof duality certifies the assignments it fixes, but returns no fixed assignments in these tests. SPVAR chooses values from samples and can exclude every optimum. Walsh elimination instead minimizes over removed spins for each retained configuration; in exact arithmetic, it preserves the minimum energy and recovers an original configuration of equal energy from every reduced one.

Supplementary Table 5: **The gain also appears with parallel tempering.** Time-to-solution index $\mathrm{TTS}_{90}^{\mathrm{pool}}$ in seconds for simulated annealing (SA) and parallel tempering (PT), with preprocessing included. The index is calculated from success fractions averaged across instances, not from individual times to solution. References are certified at $\Delta = 3$, $N \leq 200$ and best-known otherwise. A dash means that no attempt succeeds, so no finite index is inferred; it does not establish an infinite true time to solution. When every attempt succeeds, the table reports one full budget T.

N	Δ	T (s)	SA original	SA reduced	PT original	PT reduced
200	3	1	23.1	0.77	27.6	1.50
200	3	5	25.8	5.00	31.0	3.58
500	3	5	–	570	–	224
200	5	1	1.28	0.74	2.77	1.48
200	5	5	5.00	5.00	7.60	4.21
500	5	1	114	21.9	–	56.4
500	5	5	–	29.9	1146	122

Supplementary Table 6: **Energies and reference hits on new zero-field spin glasses.** The equal-sample comparison uses the same instances for each method. For each instance, ΔE_{pair} is the best energy with preprocessing minus the best energy without it. W/T/L counts wins, ties, and losses; differences within $\pm 10^{-4}$ are ties. This energy comparison is the primary result and needs no reference minimum. The secondary hit rate uses best-known references fixed before these runs, not certified optima. Roof duality fixes no spins, so its wins and losses compare searches on the unchanged Hamiltonian with different random seeds. Sample-based fixing can exclude every optimum; exact elimination preserves the minimum and allows recovery in exact arithmetic.

Cell	Method	Fixed or removed	ΔE_{pair} median	W/T/L	Hit rate	Guarantee
N=200, Δ=3	Baseline SA	0.000	–	–	0.60	–
N=200, Δ=3	Roof duality (weak)	0.000	0.000	5/11/4	0.70	exact after validation
N=200, Δ=3	Multi-start SPVAR (full)	0.070	0.000	6/11/3	0.70	none
N=200, Δ=3	Walsh elimination	0.400	0.000	8/12/0	1.00	exact + recovery
N=200, Δ=5	Baseline SA	0.000	–	–	1.00	–
N=200, Δ=5	Roof duality (weak)	0.000	0.000	0/20/0	1.00	exact after validation
N=200, Δ=5	Multi-start SPVAR (full)	0.422	0.000	0/20/0	1.00	none
N=200, Δ=5	Walsh elimination	0.300	0.000	0/20/0	1.00	exact + recovery
N=500, Δ=3	Baseline SA	0.000	–	–	0.00	–
N=500, Δ=3	Roof duality (weak)	0.000	0.516	4/0/6	0.00	exact after validation
N=500, Δ=3	Multi-start SPVAR (full)	0.020	-0.127	5/0/5	0.00	none
N=500, Δ=3	Walsh elimination	0.400	-5.291	10/0/0	0.30	exact + recovery
N=500, Δ=5	Baseline SA	0.000	–	–	0.20	–
N=500, Δ=5	Roof duality (weak)	0.000	-0.047	5/0/5	0.00	exact after validation
N=500, Δ=5	Multi-start SPVAR (full)	0.032	-0.324	6/2/2	0.30	none
N=500, Δ=5	Walsh elimination	0.306	-1.447	8/2/0	0.70	exact + recovery



Supplementary Figure 3: **Energy improvements shown separately for each setting.** These panels separate the results in main-text Fig. 3 by (N, Δ). For each instance, $\Delta E_{\text{pair}} = E_{\text{method}} - E_{\text{baseline}}$ subtracts the best energy without preprocessing from the best energy with preprocessing. Negative values favour preprocessing; no reference minimum is needed. SPVAR is sample-persistence variable reduction with multiple starts. **a–c**, Median differences at equal samples, $T = 20$ s, and $T = 60$ s. Intervals are 95% confidence intervals from paired bootstrap resampling over instances (10,000 resamples). Each $N = 200$ setting contains 20 instances and each $N = 500$ setting contains 10. The horizontal line marks zero; an open circle at zero means both the median and interval width are zero. For Walsh elimination at $N = 500$, $\Delta = 3$, the labels below the bars are -5.3, -7.6, and -6.3. At equal samples, the value is -5.291 with interval $[-6.65, -4.27]$. The 20 s and 60 s tests reuse the same seed streams and are not independent repetitions. **d**, Wins, ties, and losses at equal samples. Within each setting, the methods are ordered as roof duality, multi-start SPVAR, and Walsh elimination. Wins use the method colour, ties are grey, and losses are hatched. Counts are printed where space permits; rows containing only ties carry the method name. Summing over settings gives the counts in main-text Fig. 3d: 14/31/15, 17/33/10, and 26/34/0, respectively. Walsh elimination has no loss in any setting. Supplementary Table 6 gives reference-hit rates.

Interpretation. "Exact after validation" means that roof-duality assignments must pass independent checks of the required fixing guarantee. The procedure returns no assignments on these instances, so the Hamiltonian is unchanged. At equal samples, multi-start SPVAR shows no detectable improvement over unpreprocessed annealing; at $T = 20$ s, it benefits from the lower cost of sampling the remaining problem. Walsh elimination records no loss on any instance at any tested budget. The 20 s and 60 s tests reuse seed streams, so they are not independent

evidence.

3.6 Variable-fixing implementation and reference checks

Zero-field multi-start SPVAR. Each of the 20 starts begins with the original Hamiltonian; no state or fixing decision is carried over from another start. Each start has 500 samples. We use the first 200 to choose relative spin signs from correlations, calculated over the lowest-energy 30%:

$$C_{ij} = \langle s_i s_j \rangle_{\text{elite}} , \tag{S12}$$

where the subscript "elite" denotes that selected low-energy subset. Reversing every spin, $\mathbf{s} \to -\mathbf{s}$, leaves C_{ij} unchanged. We keep pairs with $|C_{ij}| = 1$, for which all selected samples agree on the relative sign. These pairs form a graph with signed edges. We discard any connected component whose edge signs contradict one another around a cycle. In the largest remaining component, we set the lowest-index spin to +1 and use the edge signs to assign the other spins. Fixing this one reference spin is exact when $h \equiv 0$, because optima occur in pairs $(\mathbf{s}^*, -\mathbf{s}^*)$. The relative signs of the other fixed spins still come from samples.

We draw the next 150 samples on the remaining problem and fix a spin only when all samples in the lowest-energy 20% agree on its value. We use the final 150 samples to search the remaining problem and return the best energy from all three stages. Thus, samples used to choose fixed values also contribute to the final result. The full procedure also applies weak roof duality, repeatedly removes spins with zero or one neighbour by exact elimination, and combines solutions from separate connected components. On this problem class, their median contributions are 0–1 spins, 0 spins, and 0 energy improvement, respectively.

Reference searches and follow-up. To set references for the 60 instances, we ran four long simulated-annealing attempts and four long parallel-tempering attempts on each of the two formulations. This gives 960 runs in total. Each run receives at least 60 s at $N = 200$ or 120 s at $N = 500$. None of these seeds is reused in the benchmark. We recovered the removed spins in every reduced state and evaluated its energy again in the original Hamiltonian; no discrepancy exceeded 10^{-8}. For each instance, the reference is the lowest energy whose stored configuration passes this check. None is a certified optimum. These references belong only to the variable-fixing comparison and are kept separate from the development-benchmark references.

A second reference search stopped after the first nine instances on a priority list. It completed 72 of the planned 480 parallel-tempering runs, each with a 300 s budget. All 72 runs reached the first-stage energy without improving it; the other 408 planned runs were not performed. Because the nine instances were not selected at random, this result does not estimate how often the same search would reproduce the references across the full set.

Which formulation found the references. Both formulations contributed candidate reference energies. In three of the four settings, the original formulation independently reached the final reference on every instance. At $N = 500$, $\Delta = 3$, it did so on only one of ten instances; the reference search reached the other nine targets only with the reduced formulation. These targets can therefore favour the reduced formulation when comparing reference-hit rates. The direct energy comparison does not need those targets: Walsh elimination wins all ten instances, with median $\Delta E_{\text{pair}} = -5.291$ and a 95% confidence interval of $[-6.65, -4.27]$ from paired bootstrap resampling.

Roof-duality validation. We tested roof duality with `dwave-preprocessing` 0.6.10. For variables labelled $0, \ldots, N-1$, the tested fast code path returns positions in the model's internal ordering instead of the variable labels. We checked the result by complete enumeration on 60 random models with nonzero local fields. Assignments satisfied strong persistency in 25% of the integer-labelled cases and in 100% after relabelling; the lower bound was correct in both cases. The reported benchmarks therefore use the relabelled path.

In tests without local fields, `strict=True` returned fixed assignments in 18 of 40 cases; 55% of the returned assignments passed the independent strong-persistency check. With `strict=False`, the tested path excluded every exact optimum in 6 of 26 tests. The benchmark code therefore checks every returned set of assignments and raises an error if a check fails. Neither mode returns any validated fixed assignment on the manuscript instances, so these defects do not affect the reported results.

3.7 Protocol C: time-budget checks on 100 separate instances

The protocol is described in Supplementary Note 3.9.

Supplementary Table 7: **Wins minus losses on a separate set of 100 instances.** Protocol C compares each method with unpreprocessed annealing on the same instance; no reference energy is needed. Each entry is wins minus losses. Parentheses for Walsh elimination give wins/ties/losses. Both roof-duality modes fix no spins, so those columns compare runs on the unchanged Hamiltonian with different random seeds. These results are separate from the 60-instance comparison and are not combined with it.

T (s)	Walsh elimination	SPVAR full	SPVAR core	Alignment only	Roof strict	Roof weak
5	+35 (35/65/0)	+32	+27	+24	+7	+2
20	+29 (29/71/0)	+24	+19	+11	+2	−1
60	+23 (23/77/0)	+4	+5	−2	−5	−4

Walsh elimination has no losses at any budget. For full SPVAR, wins minus losses are +32 at $T = 5$ s and +24 at $T = 20$ s, but only +4 at $T = 60$ s. For comparison, searches on the unchanged Hamiltonian after roof duality give values from -5 to +7 across these budgets. SPVAR's 60 s result lies within that range. As in the 60-instance comparison, SPVAR helps most under tight time limits.

3.8 Quadratic benchmark protocol

Instances. We use random Δ-regular graphs G with edge set $E(G)$, zero local fields, and the Gaussian spin-glass Hamiltonian

$$H(\mathbf{s}) = \sum_{(i,j)\in E(G)} J_{ij} s_i s_j, \qquad J_{ij} \sim \mathcal{N}(0,1). \tag{S13}$$

Each setting fixes (N, Δ), with $N \in \{100, 200, 500, 1000\}$ and $\Delta \in \{3, 5\}$. There are $M = 20$ instances per setting at $N \le 200$, $M = 10$ at $N = 500$, and $M = 5$ at $N = 1000$.

Solver, time budgets, and preprocessing. OpenJij simulated annealing uses 1000 sweeps per returned sample. For each instance, formulation, solver, and budget, we run 10 independent attempts, with $T \in \{0.5, 1, 2, 5\}$ s per attempt. Both formulations use the same seeds and

`SASampler().sample_hubo`. An attempt can return multiple samples. Each reduced attempt receives $T - t_{\mathrm{prep}}$ of solver time, so the full preprocessing time t_{prep} counts against its budget. The OpenJij version was not recorded for these development runs. The variable-fixing comparison (Supplementary Note 3.9), the higher-order benchmark (Supplementary Note 4.4), the held-out study (Supplementary Note 6.1), and the solver cross-check used OpenJij 0.11.6.

By default, the library calculates the inverse-temperature limits $\beta_{\min}$ and $\beta_{\max}$ from the submitted coefficients. The two formulations therefore receive different annealing schedules on 19 of the 20 instances in the certified quadratic setting. A rerun on a second machine gives an original-to-reduced time-to-solution index ratio of $19.8\times$ $[7.3, 41.9]$ with these separate schedules. Using the original Hamiltonian to set one schedule for both formulations gives $21.8\times$ $[10.3, 56.2]$; using the reduced Hamiltonian to set their shared schedule gives $26.5\times$ $[10.3, 54.9]$. These indices use success fractions averaged across instances. The intervals overlap, and either shared schedule retains the gain.

Elimination schedule. For both graph degrees, a prior parameter sweep selected the target fraction $f = 0.4$, the cap $d_{\mathrm{cap}} = \Delta + 1$, and `min_degree` ordering. Here $\widetilde{d}_a$ counts neighbours in the stored fill graph, which retains edges even when their weights cancel to zero. Ordering and applying the cap to this count therefore bound the actual neighbour count of each spin selected for removal (main-text Section 2.2). Supplementary Note 3 also gives results for $f = 0.2$.

Reference energies. At $\Delta = 3$ and $N \le 200$, $E_{\mathrm{ref}}^{\mathrm{main}}$ is the certified minimum. We use GenericTensorNetworks.jl to combine local energy tables and minimize over their shared spins, a calculation called min-plus (tropical) tensor contraction. The algorithm is exact in exact arithmetic; our calculation uses double-precision floating-point arithmetic (binary64) on the stored coefficients. We independently evaluate the returned configuration in the original Hamiltonian.

The order of contraction controls the memory needed. A plan with space complexity sc has a largest intermediate tensor of 2^{sc} elements. TreeSA chooses plans stochastically, and repeated plans for one instance can differ by two or three in sc. Early certifications used one TreeSA plan; later settings and all retries selected the best of ten TreeSA plans and the library's greedy default. We record both sc and measured peak memory, because the largest tensor alone does not determine total memory use. The certifications required 4.3–37.5 GB at sc = 24–29, with a longest single solve of 3798 s and no failure within the machine's 48 GB. All 40 instances in the two settings were certified, and each computed minimum agreed with its stored reference to the table precision of 5×10^{-7}. These are numerical verifications in binary64 arithmetic, as in Supplementary Note 6.1. Across these quadratic settings and the certified higher-order settings, certification confirmed the reference energy on all 60 instances that could be checked. It does not establish optimality for the remaining best-known references.

For the other settings, even the best contraction orders found require more memory than is available: sc = 62–69 at $\Delta = 5$, $N = 200$; 66–78 at $\Delta = 3$, $N = 500$; 154 at $\Delta = 3$, $N = 1000$; and 180–183 at $\Delta = 5$, $N = 500$. These values make direct contraction infeasible with the available memory and the orders found. For these settings, $E_{\mathrm{ref}}^{\mathrm{main}}$ is the lowest energy found across long simulated-annealing runs, at least six independent 60 s parallel-tempering runs on each formulation, and all benchmark runs. An independent implementation evaluates every state in the original Hamiltonian after the removed spins have been recovered. Because the benchmark runs contributed to these references, the comparison measures closeness to the best energies found, rather than success against targets fixed independently before testing. At $N = 1000$, longer search still lowered some references, so we report only the absolute gap

$$\delta E_{\mathrm{main}} = E_{\mathrm{best}} - E_{\mathrm{ref}}^{\mathrm{main}}. \tag{S14}$$

Success fractions and restart estimates. An attempt succeeds when $E_{\mathrm{best}} \le E_{\mathrm{ref}}^{\mathrm{main}} + 10^{-4}$. For instance i, let p_i be the observed fraction of successful attempts. Their mean across the M instances is the pooled success fraction, $\bar{p}_{\mathrm{succ}} = M^{-1} \sum_{i=1}^{M} p_i$. The time-to-solution index in main-text Eq. (22) inserts this mean into the restart formula as though every instance had that success probability. It allows a noninteger number of attempts, so it can be shorter than one attempt. It is neither the mean nor the median time to solve individual instances.

For an integer number of attempts $n \ge 1$, main-text Eq. (23) first estimates success on each instance and then averages. Since $1 - (1 - p)^n$ is concave in p, applying the formula to the mean fraction can overestimate average success when instances differ in difficulty. Rounding the pooled estimate gives

$$n_{90}^{\mathrm{pool}} = \left\lceil \frac{\ln(0.1)}{\ln(1 - \bar{p}_{\mathrm{succ}})} \right\rceil, \qquad t_{90}^{\mathrm{pool}} = T n_{90}^{\mathrm{pool}}, \tag{S15}$$

but rounding does not give a 90% success guarantee for each instance or for the average over the fixed test set. When every recorded attempt succeeds ($\bar{p}_{\mathrm{succ}} = 1$), we display one full budget T; this is a plotting convention, not an upper confidence bound. When no attempt succeeds, we infer no finite index. In the certified quadratic setting, 13 of 20 instances have no success in ten attempts of the original formulation. Those observed fractions cannot give finite restart estimates for the individual instances, but they do not establish that the true probabilities are zero.

3.9 Variable-fixing comparison protocol

Protocols. We compare Walsh elimination, roof duality, and the published SPVAR procedure for zero-field models with multiple starts. The test uses 60 instances generated with new seeds, with $N \in \{200, 500\}$ and $\Delta \in \{3, 5\}$: 20 instances per setting at $N = 200$ and 10 at $N = 500$. The baseline applies the same simulated-annealing sampler to the unchanged Hamiltonian. The equal-sample protocol gives every method 20 independent starts of 500 samples per instance. Two wall-clock protocols instead give every method the same total budget, $T = 20$ s or $T = 60$ s, across the 20 starts. The two timed protocols reuse the same seed streams, so they are not independent repetitions. Their reference energies $E_{\mathrm{ref}}^{\mathrm{conf}}$ come from additional searches with separate seeds and are fixed before the comparison. They are best-known energies, used only for the secondary reference-hit rates. All instances, solver runs, and references were produced in one environment.

Published zero-field multi-start SPVAR. We follow the zero-field workflow of Karimi *et al.* [4]. Each of 20 independent starts begins from the original Hamiltonian and divides 500 samples in the prescribed $40\%/30\%/30\%$ ratio: first to fix relative signs from correlations, then to fix values that agree across the selected samples, and finally to search the remaining problem. The procedure also uses weak roof duality, exact elimination of spins with zero or one neighbour, and recombination of solutions from separate connected components. Supplementary Note 3.6 gives the complete rules and checks.

Reference searches and their role. We constructed the 60 confirmation references from 960 long simulated-annealing and parallel-tempering runs across the original and reduced formulations, using seeds disjoint from the benchmark. For every reduced state, the removed spins were recovered and the energy was evaluated again in the original Hamiltonian; all runs passed the 10^{-8} check. A follow-up of 72 parallel-tempering runs, each lasting 300 s, found no

lower energy on nine priority instances. Those instances were not selected at random, so this does not estimate how often the references would be reproduced across the full set. At $N = 500$, $\Delta = 3$, nine of ten references were reached during the reference search only by the reduced formulation. We therefore use the direct comparison of achieved energies as the primary measure and reference-hit rates as secondary. Supplementary Note 3.6 gives the full reference history.

Roof-duality implementation. We first relabel the variables. We then apply roof duality with `dwave-preprocessing` 0.6.10. Every returned assignment is independently checked for the required fixing guarantee because the tested library paths have the two label-handling defects described in Supplementary Note 3.6. Neither strict nor weak mode returns any validated fixed assignment on the manuscript instances. The subsequent annealing therefore uses the unchanged Hamiltonian, so its wins and losses measure differences between searches without spin removal.

Protocol C. Protocol C compares achieved energies on 100 separate instances, without requiring reference minima. It uses 20 starts per method and budgets $T \in \{5, 20, 60\}$ s. Supplementary Table 7 gives the results; they are not combined with the 60-instance comparison. An older sampling path was retained only for diagnostics and contributes no result to the manuscript.

Supplementary Note 4 Higher-order benchmark checks

These checks test when the interactions created by elimination become too costly, whether changing the time budget changes the result, and whether a second solver gives a similar gain. Supplementary Note 4.4 gives the solver checks and time accounting.

4.1 When preprocessing uses the available budget

Supplementary Table 8: **When preprocessing outweighs the reduction in spin count.** The default cap limits the number of terms containing the selected spin. An exact interaction-order cap instead checks the terms actually created by a trial Walsh transform. The table distinguishes the cost of building the reduced model from its success once built. All time-budget comparisons include preprocessing.

Problem	Schedule	Observed outcome
$k = 3, \Delta = 3$	Default incident-term cap	Median preprocessing is below 2 s at $N = 512$; the gain includes this cost.
$k = 3, \Delta = 3$	Exact interaction-order cap with `min_degree` or `min_fill`	Trial Walsh expansions raise preprocessing to 11–25 s at large N. Short budgets are then lost to preprocessing.
$k = 4, \Delta = 3$	Paper-default elimination schedule	Preprocessing takes 108–170 s in the two tested cells. Once the reduced model is built, the observed success probability at $N = 64$ is 0.92 for the reduced model against 0.79 for the original. Including preprocessing favours the original.
Dense LABS	Practical interaction-order caps	No spin is eligible. The compiler returns the original Hamiltonian.

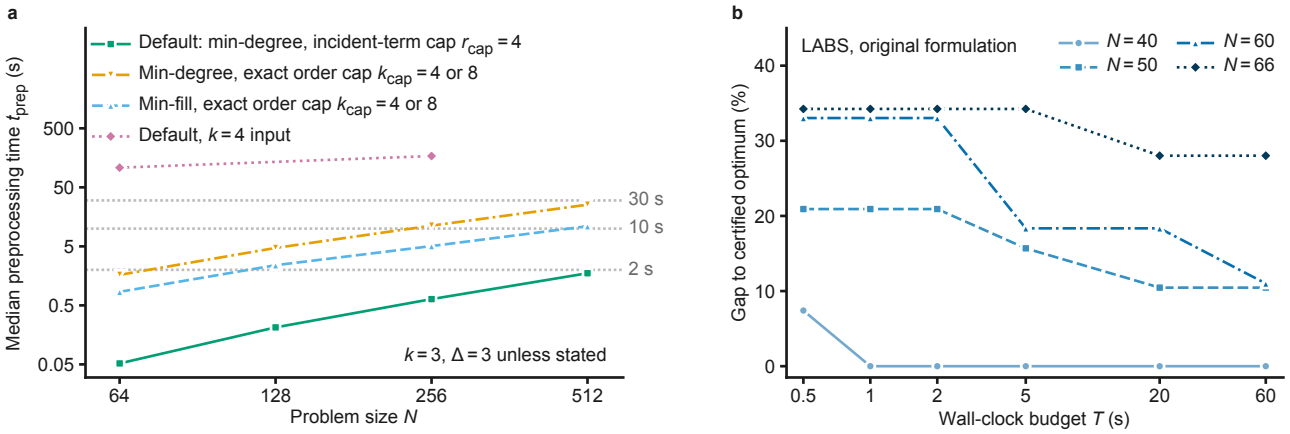
The default schedule checks local counts before attempting a transform. Checking the actual order of the terms created requires trial transforms; at $N = 512$, those checks use 11–25 s of a 30 s budget.

4.2 Changing the budget while keeping the same instances

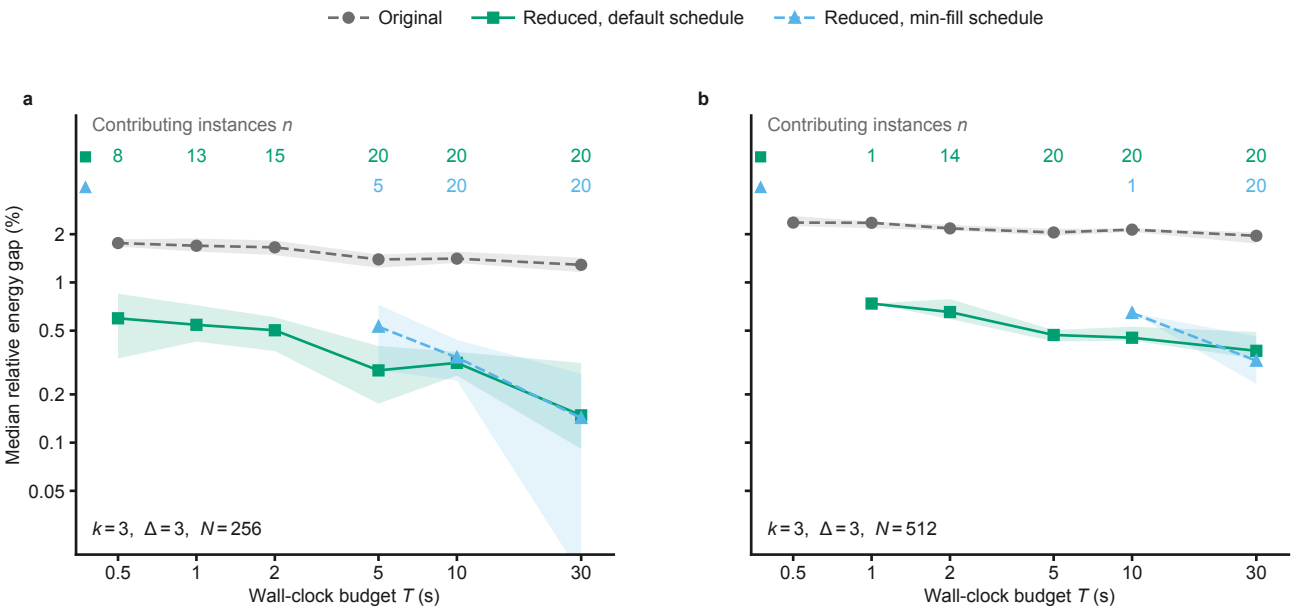
At short budgets, preprocessing can leave no time to solve some instances. A curve that omits those instances can therefore change its sample as the budget grows. Supplementary Fig. 5 shows these changing sets. Supplementary Fig. 6 repeats the default-schedule comparison using the same instances at every budget from $T = 2$ s onward.

4.3 Check with a second solver

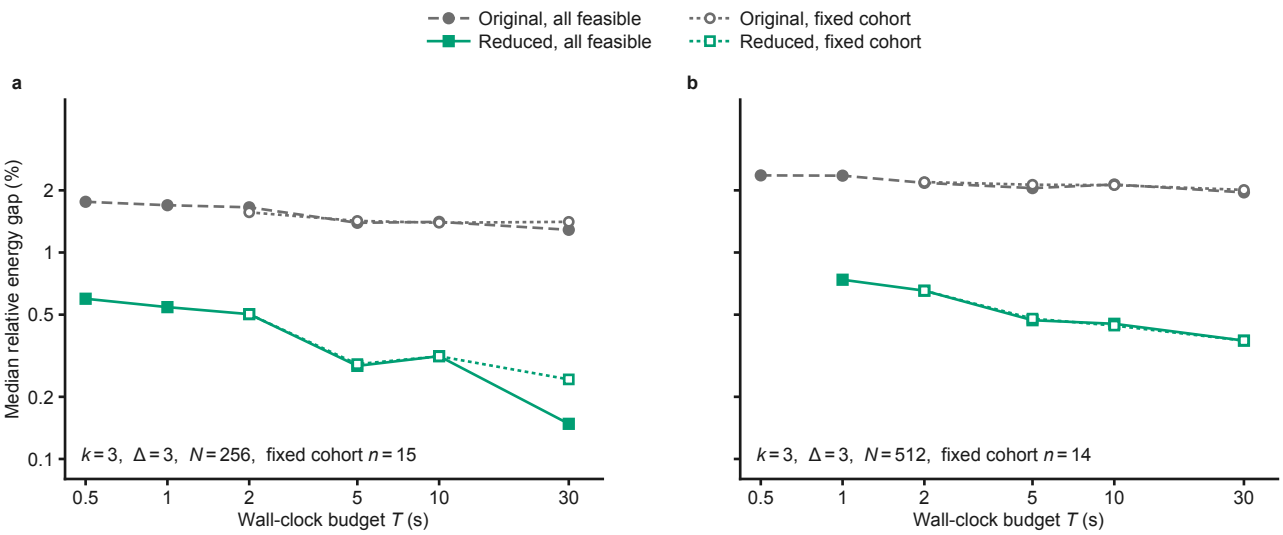
The cross-check protocol is described in Supplementary Note 4.4.



Supplementary Figure 4: **Preprocessing can use the entire time budget. a**, Median preprocessing time t_{prep} against N for $k = 3$, $\Delta = 3$, with logarithmic axes; the default schedule at $k = 4$ is also shown. The default uses `min_degree`, targets $f = 0.4$, and accepts only spins in at most $r_{\mathrm{cap}} = \Delta + 1 = 4$ terms, without trial transforms. The other two series use `min_degree` or `min_fill` and check the actual order of the output through trial transforms. Each of their points combines the four tested settings: order caps 4 and 8 and target fractions 0.2 and 0.4. There are 20 instances at $k = 3$, giving 20 runs per default point and 80 per combined point, and 10 instances at $k = 4$. At $N = 512$, preprocessing takes 11–25 s with the output-order checks but a median of 1.8 s with the default cap. Dotted lines mark budgets of 2, 10, and 30 s. For $k = 4$, preprocessing takes 108–170 s in the two tested settings, exceeding every budget tested. **b**, Gap to the certified optimum against time budget for low-autocorrelation binary sequence (LABS) problems, with one deterministic instance at each length N. Only $N = 40$ reaches the optimum within the tested budgets. No spin can be removed under the tested order caps, so the original and reduced Hamiltonians are identical.



Supplementary Figure 5: **Energy gaps decrease as the time budget grows.** Median relative gap to the reference for $k = 3$, $\Delta = 3$ at **a**, $N = 256$, and **b**, $N = 512$. Both axes are logarithmic. Lines show medians across the contributing instances; bands are percentile-bootstrap 95% intervals from 5000 resamples, calculated separately for each curve and budget. The default schedule uses `min_degree` with $r_{\text{cap}} = \Delta + 1$. The min-fill schedule checks the actual output order through trial transforms. Each reduced attempt includes preprocessing. Instances for which preprocessing uses the full budget are omitted from the reduced curves, while the original curves use all 20 instances at every budget. Counts n above the panels identify the contributing instances for each reduced series by colour and marker. The min-fill series starts at $T = 5$ s in **a** and $T = 10$ s in **b**; the default series has no point at $T = 0.5$ s in **b**. At $N = 512$, the first point of each reduced series uses only one instance, so no interval for variation across instances is drawn. The reduced median is lower at every plotted budget, but when $n < 20$ it is based on a smaller set than the original median. Supplementary Fig. 6 compares the default schedule on the same instances from $T = 2$ s onward.



Supplementary Figure 6: **The lower gaps persist when the same instances are compared.** The default-schedule comparison of Supplementary Fig. 5 is repeated on the instances for which preprocessing leaves solver time at $T_0 = 2$ s: $n = 15$ at $N = 256$ and $n = 14$ at $N = 512$. This is the shortest budget that includes most of each setting. For every $T \geq T_0$, both formulations use this same selected set. **a**, $N = 256$. **b**, $N = 512$. Filled markers show the budget-dependent sets in Supplementary Fig. 5; open markers and dotted lines show the fixed sets. Both axes are logarithmic. Fixing the set changes some medians, but the reduced formulation remains closer to the reference at every budget shown for that set.

Supplementary Table 9: **A second higher-order solver also benefits from elimination.** At $T = 2$ s, both solvers use the same 20 instances, certified minimum energies, and seed schedule, with preprocessing included. The columns p_{orig} and p_{red} give observed success fractions averaged across instances. The TTS columns are indices calculated from those averages, not measured times for each instance. The ratio is $(\text{TTS}_{90}^{\text{pool}})^{\text{orig}}/(\text{TTS}_{90}^{\text{pool}})^{\text{red}}$, with 95% confidence intervals from paired bootstrap resampling over instances.

Solver	p_{orig}	p_{red}	TTS original (s)	TTS reduced (s)	Ratio
OpenJij 0.11.6	0.220	0.950	18.53	1.54	12.1 [7.6, 22.4]
`qubovert` 1.2.5	0.150	0.800	28.34	2.86	9.9 [6.4, 17.8]

Both solvers favour elimination: their intervals overlap, and both lower bounds exceed 3×. Elimination raises the sample rate by 1.24× under `qubovert` and 1.46× under OpenJij. These measurements do not separate the effect of returning more samples from changes in success per sample. The gain can therefore differ between implementations.

4.4 Higher-order benchmark protocol

Solver and time accounting. OpenJij `sample_hubo` uses spin variables and 1000 sweeps per returned sample. We checked that it introduces no auxiliary variables and that run time changes with the number of higher-order terms supplied. Seeds and budgets follow Supplementary Note 3.8. Each reduced attempt receives $T - t_{\text{prep}}$ of solver time; the full preprocessing cost is included in every attempt.

Problem families. Family A consists of sparse random higher-order spin glasses,

$$H(\mathbf{s}) = \sum_{e \in \mathcal{E}} J_e \prod_{i \in e} s_i, \qquad J_e \sim \mathcal{N}(0,1), \tag{S16}$$

on k-uniform, Δ-regular hypergraphs, where each interaction e contains exactly k spins and each spin occurs in exactly Δ interactions. The main tests use $k = 3$ and $\Delta \in \{3, 5\}$ at four sizes up to $N = 512$ for $\Delta = 3$ and $N = 513$ for $\Delta = 5$, with 20 instances per setting. A four-body test uses $k = 4$, $\Delta = 3$, and $N \in \{64, 256\}$, with 10 instances per setting. Family B consists of planted 3-regular 3-XORSAT problems at four sizes up to $N = 128$, with 20 instances per size. Each 3-XORSAT instance consists of parity constraints, each on three spins, and has a minimum energy of zero by construction. We verify this by elimination using arithmetic modulo two, $\mathrm{GF}(2)$. Although this solves 3-XORSAT in polynomial time, planted 3-regular instances are an established benchmark for local search and Ising machines because these methods scale poorly on them [14]. Family C contains LABS problems, which minimize correlations of a binary sequence with shifted copies of itself, excluding zero shift. Their Hamiltonians are dense and contain terms of order up to four. We use one deterministic instance per length and the published exact energies for $N \leq 66$ [15], reproducing the $N = 20$ value by complete enumeration.

References. Of the 266 instances, 136 have certified minimum energies: 50 from tensor-network contraction, 80 from $\mathrm{GF}(2)$ elimination, and 6 from published LABS results. The other 130 use best-known references $E_{\mathrm{ref}}^{\mathrm{HO}}$ from the additional searches in Supplementary Note 3.8. Every reported setting uses the same reference type for all its instances: certified or best-known.

At $k = 3$, $\Delta = 3$, $N = 64$, contraction takes a median of 8.0 s per instance, with a range of 6.0–47.4 s. At $k = 3$, $\Delta = 5$, $N = 63$, the median is 59 s, the contraction orders have space complexity 27–30, and peak memory reaches 39.0 GB. Certifying all 20 instances took 51 min. For the 3-XORSAT instances, $\mathrm{GF}(2)$ elimination takes 0.12–0.66 ms.

The largest completed higher-order contraction had space complexity 30, took 1308 s, and used a peak memory of 39.0 GB. The setting $k = 3$, $\Delta = 3$, $N = 128$ retains best-known references. So do $k = 3$, $\Delta = 3$, $N = 256$ and $k = 4$, $\Delta = 3$, $N = 256$, whose best contraction orders have space complexities 61 and 81. Certified minima let us compare the formulations against the same verified target. They do not establish an advantage over exact solvers: contraction itself solves the $N = 64$ instances in seconds. The larger-instance comparisons measure closeness to best-known energies, not to certified minima.

Schedule. The default elimination schedule targets $f = 0.4$ using `min_degree` and the rule $r_a \leq r_{\mathrm{cap}} = \Delta + 1$ from main-text Section 2.2. This limits the number of terms containing the selected spin, not the number of its neighbours. By Eq. (16), a new term can involve up to $(k-1)\, r_{\mathrm{cap}}$ spins, and this bound grows as the current order k grows. We therefore record both neighbour count and output order. A separate limit of 16 neighbours in the stored graph bounds the transform size. When a cap on the actual output order is used, candidates whose neighbour count exceeds that order cap require trial expansions. Such candidates are rejected without a trial when the stored neighbour count exceeds 12. In the certified setting $k = 3$, $\Delta = 3$, $N = 64$, the schedule reduces 64 spins to 38 and the 64 original terms to 42–59 terms, with induced order $k_{\max} = 6$–10 across the 20 instances. Every accepted spin in this setting had $r_a \leq 4$, yet its neighbour count d_a ranged from 4 to 11. A neighbour cap of four would therefore have rejected 516 of the 520 accepted steps.

Choices fixed before the benchmark. Before the benchmark runs, we fixed the problem families, settings, success criterion, and comparison rules. Settings with certified minima use 25 attempts per instance, formulation, and budget when $T < 10$ s; the repository manifest gives the complete attempt schedule. We also fixed a $3\times$ threshold for the original-to-reduced pooled index ratio when reporting a clear gain. Before the benchmark, a pilot checked the stated adoption criterion for Family A on 10 of the 20 instances at $k = 3$, $\Delta = 3$, $N = 64$. The pilot chose no parameter: the target fraction and numerical cap matched the quadratic default, but the cap counted terms containing the selected spin, as described above.

Certification after the runs and complete reporting. After the benchmark runs, tensor-network contraction with repeated planning certified one additional setting. This changed only the reference status, not the attempts, seeds, budgets, schedules, or solver calls. Both formulations use identical seed sets. We report every setting with certified minima, including results below the pre-specified $3\times$ threshold and results with no observed success in one formulation, for which no finite ratio is inferred. Supplementary Note 4 reports alternative orders, caps on the actual output order, and target elimination fractions $f \in \{0.2, 0.4\}$.

Statistics. Confidence intervals in the text and tables use paired bootstrap resampling over instances with $10{,}000$ resamples. The display bands in Supplementary Fig. 5 instead show percentile-bootstrap intervals for each median, with 5000 resamples drawn separately from the contributing instances for each curve and budget. Resampling uses one random stream across settings in a fixed order; reproducing an interval therefore requires replaying that stream. The pre-specified statistic is the index calculated from mean success fractions. Ratios calculated separately for individual instances also favour elimination where they can be estimated. At $k = 3$, $\Delta = 3$, $N = 64$, their median is $17.2\times$ $[9.5, 30.2]$, compared with $12.6\times$ for the pooled index ratio.

Check with a second solver. We repeated the certified setting $k = 3$, $\Delta = 3$, $N = 64$ with `qubovert` 1.2.5 and OpenJij on the same machine, passing the original and reduced polynomials directly to both solvers. We checked that neither solver call converts them to quadratic models. Reported energies agree with an independent evaluator to within 2×10^{-14}. The faster machine returns higher observed success fractions at the same T than the development runs, but the index ratios remain similar (Supplementary Table 9).

Supplementary Note 5 Additional algebraic examples

These examples check elimination by enumerating the spin configurations; they do not measure solver speed. The J-Möbius ladder gives an explicit reduced Hamiltonian, the two-bit adder gives every local polynomial, and the factorization examples check recovery from small reduced models.

5.1 J-Möbius ladder

For N divisible by four, the antiferromagnetic ladder is

$$H_M(\mathbf{s}) = \sum_{i=1}^{N} s_i s_{i+1} + J \sum_{i=1}^{N/2} s_i s_{i+N/2}, \qquad s_{N+1} \equiv s_1, \tag{S17}$$

with $J \geq 0$ [16, 17]. The state $\mathbf{S}_0$ alternates around the ring, satisfies every ring bond, and violates every cross-ring bond, with energy $-N+NJ/2$. The competing class $\mathbf{S}_1$ satisfies every cross-ring bond but has parallel spins on two ring bonds at opposite positions, with energy $-N+4-NJ/2$. The energies cross at

$$J_c = \frac{4}{N}. \tag{S18}$$

For $N = 8$, eliminating $s_1, s_3, s_6, s_8, s_2, s_4$ gives

$$H_M^{[6]} = \begin{cases} -6 + 2(2J-1)s_5 s_7, & J < \dfrac{1}{2}, \\ -4 - 4J, & J \geq \dfrac{1}{2}. \end{cases} \tag{S19}$$

For $J < 1/2$, the minimum requires $s_5 s_7 = +1$, and reverse substitution recovers $\mathbf{S}_0$ and its global flip. For $J > 1/2$, the reduced energy is independent of s_5 and s_7, and reverse substitution through the tie branches of Eq. (7) recovers the degenerate $\mathbf{S}_1$ states. The two classes have the same minimum energy at $J = 1/2$. Complete enumeration verifies Eq. (S19) for every assignment of (s_5, s_7) and reproduces $J_c = 1/2$. This checks the reduced energy, not how a solver crosses energy barriers.

5.2 Two-bit adder

We encode a two-bit ripple-carry adder with input spins $s_1, \ldots, s_4$, an intermediate carry s_5, output spins s_6, s_7, s_8, and a fixed input carry $s_9 = -1$. Define the spin carry function

$$C(x, y, z) = \frac{1}{2}\left(x + y + z - xyz\right). \tag{S20}$$

A constraint Hamiltonian is

$$\begin{aligned} H_a^{[0]} = {} & \left(s_6 - s_2 s_4 s_9\right)^2 + \left[s_5 - C(s_2, s_4, s_9)\right]^2 \\ & + \left(s_7 - s_1 s_3 s_5\right)^2 + \left[s_8 - C(s_1, s_3, s_5)\right]^2. \end{aligned} \tag{S21}$$

With bit value $b = (s+1)/2$, s_2, s_4 are the low-order input bits, s_1, s_3 are the high-order input bits, and s_6, s_7, s_8 are the sum bits from low to high order. The Hamiltonian has minimum energy zero, attained by 16 ground states, one for each input assignment. Eliminating s_1, s_2, s_3, and s_4 in that order gives the following local polynomial at each step, with all preceding eliminations included:

$$\begin{aligned} P_1 &= -2s_3 s_5 s_7 - s_8 + s_3 s_5 s_8, & P_2 &= -s_5 - s_4 s_5 + 2s_4 s_6, \\ P_3 &= -s_8 - s_5 s_7 s_8, & P_4 &= -s_5 + s_5 s_6. \end{aligned} \tag{S22}$$

At step j, the compiler replaces $s_j P_j$ by $-|P_j|$ and stores the reverse rule. The final Hamiltonian is

$$H_a^{[4]} = 2 + s_6 + s_5\left(1 + s_6 - s_7 - s_8\right) + s_7 s_8. \tag{S23}$$

The reduced Hamiltonian has minimum energy zero at nine of the sixteen configurations of $(s_5, \ldots, s_8)$. Enumeration confirms that these are exactly the configurations left when the input spins are removed from the sixteen valid additions. Recovering the input spins and keeping both signs at every tie recovers all sixteen original ground states.

5.3 Factorization encodings

The three-spin encoding for $291311 = 523 \times 557$ is

$$H_{3p} = \frac{3}{2} + \frac{1}{2}\left(s_1 s_2 + s_1 s_3 - s_2 s_3\right) \tag{S24}$$

[18]. Eliminating s_1 gives

$$H_{2p} = 1 - s_2 s_3, \tag{S25}$$

whose two minima recover the two possible orders of the factors. This particular encoding has already been simplified. Its three-spin size does not bound the resources needed to factor an arbitrary integer of the same length.

For the ten-spin diagonal encoding of Ref. [19], eight eliminations give

$$H_c^{[8]} = -425 - 61 s_{10} - 15 s_9 + 33 s_{10} s_9. \tag{S26}$$

Its unique minimum is

$$s_9 = -1, \qquad s_{10} = 1, \qquad H_c^{[8]} = -504, \tag{S27}$$

and reverse substitution gives

$$(s_1, \ldots, s_{10}) = (1, -1, 1, 1, 1, -1, 1, 1, -1, 1). \tag{S28}$$

For every retained configuration, complete enumeration verifies that the reduced energy equals the minimum over the removed spins and that recovery attains that energy. These checks concern the stated small encodings; they do not reduce the general complexity of integer factorization.

Supplementary Note 6 Confirmation on new instances: full statistics

Supplementary Table 10 reports all measures from the confirmation test on new instances (main-text Section 2.5): success fractions, achieved energies, restart estimates, and the pooled indices used for comparison with the development tests.

The protocol, analysis code, and confirmation criterion were recorded before instance generation. Their recorded SHA-256 prefix is `6664f0fd1aa5`; the repository manifest contains the full digests. The protocol commit is `f65b4f41`, and the results commit is `fad0b091` (Supplementary Note 6.1).

6.1 Protocol fixed before the new instances were generated

Before drawing the 256-bit master seed, we recorded the protocol, analysis code, and confirmation criterion in version control and saved their hashes. Supplementary Note 6 gives the hash prefix and commit identifiers; the repository manifest contains the full digests. The existing production data were unchanged. The analysis was fixed before any result from the new instances was examined, and no model, subset, or outcome measure was added later.

Supplementary Table 10: **Success and energy gains under a protocol fixed in advance.** The test starts with 40 new quadratic and 40 new three-body instances. One quadratic reference lacks its required second verification, so success-based rows use 39 quadratic instances; energy rows use all 40. All three-body rows use 40 instances. For each formulation, $\bar{p}$ is the observed success fraction averaged across instances. The increase Δp is reduced minus original, with a 95% interval from bootstrap resampling of instances and then paired attempts. W/T/L counts instances where the reduced formulation does better, equally, or worse. For energy, each attempt contributes the lowest energy among its returned samples, evaluated after recovery in the original Hamiltonian. We first take the median of the 50 paired reduced-minus-original attempt differences on each instance, then take the median across instances. This is ΔE (main-text Eq. (24)); its 95% interval comes from the same bootstrap of instances and then paired attempts. The energy comparison needs no reference minimum; its W/T/L counts classify the instance medians with tolerance 10^{-4}, and the sign test uses the non-tied instances. The restart count n_{90} is the first integer at which estimated success, averaged across instances, reaches 0.9. The original quadratic curve approaches only $33/39 \simeq 0.846$ because 6 of 39 observed success fractions are zero, so it has no such crossing. This does not establish zero true probabilities or an infinite true restart count. The pooled index ratio is a separate summary calculated from mean success fractions, not the confirmation criterion.

	$\Delta = 3$, $N = 200$, $T = 1$ s	$k = 3$, $\Delta = 3$, $N = 64$, $T = 2$ s
$\bar{p}$ original	0.104	0.169
$\bar{p}$ reduced	0.975	0.875
Δp [95%]	$+0.871$ $[0.819, 0.912]$	$+0.707$ $[0.655, 0.755]$
W/T/L, success	39/0/0	40/0/0
ΔE [95%]	-0.730 $[-0.895, -0.536]$	-0.126 $[-0.161, -0.080]$
W/T/L, energy	39/1/0	40/0/0
n_{90} original → reduced	undefined → 1	22 → 2
Pooled index ratio (descriptive)	33.7×	11.3×

Instances and references. We generated 40 instances in each setting from the master seed using networkx 3.6.1 and saved them once. Every later stage loads these files and checks their hashes. Setting Q uses connected random graphs with three neighbours per spin, no local fields, and Gaussian couplings at $N = 200$. Setting H uses random Gaussian spin glasses with three spins per interaction and three interactions per spin at $N = 64$. After modification to accept a supplied seed, the generator reproduced all 20 production instances exactly. We calculate reference minima by tropical tensor contraction with GenericTensorNetworks.jl (`SingleConfigMin`). Two calculations with independently planned contraction orders must agree, and an independent evaluator checks the returned state in the original Hamiltonian. These are checks in double-precision floating-point arithmetic (binary64), not exact-arithmetic certificates. Instance 36 in setting Q (its zero-based index in the saved data) completed its first calculation (sc = 27, 504 s), but the second exceeded 7200 s. The planner, fixed in advance, minimizes space complexity rather than run time. This instance therefore lacks the two verifications required for an accepted reference, but remains in the energy comparison, which needs no known minimum.

Solver and schedule. We use `SASampler().sample_hubo` for both formulations, with 1000 sweeps per returned sample. For each instance, the library's automatic rule is applied to the original Hamiltonian, and the resulting inverse-temperature range is supplied explicitly to both formulations. Because this rule is driven by the smallest coupling, $\beta_{\max}$ spans 1.1×10^2 to

9.1×10^4 across the quadratic instances (median 8.8×10^2), with both formulations using the same value for each instance.

Each instance receives 50 attempts per formulation, paired by the same attempt seed, with budgets of T = 1 s (Q) and T = 2 s (H). Every reduced attempt includes the full preprocessing time. If preprocessing used the whole budget, the attempt would count as a failure; this did not occur. The elimination settings are the development defaults, fixed with the protocol: $f = 0.4$ and `min_degree`, with d_{cap} = 4 for Q and r_{cap} = 4 for H. For H, the separate limit of 16 neighbours in the stored graph also remains in place.

The reductions were $200 \to 120$ and $64 \to 38$ spins. The largest new terms contained 4 and 12 spins, respectively. Reconstructing the local energy tables from the retained Walsh coefficients gave discrepancies of at most 1.8×10^{-15} per step and 7.1×10^{-14} cumulatively.

Outcome measures and statistical analysis. For each instance, we subtract the original observed success fraction from the reduced fraction, then average these differences. This is the primary measure, $\Delta p = \bar{p}_{\mathrm{red}} - \bar{p}_{\mathrm{orig}}$. We use $20{,}000$ bootstrap resamples, first resampling instances and then paired attempts within each instance. A setting confirms the gain when the entire two-sided 95% interval lies above zero. Instances with no observed successes remain in the analysis.

For the secondary energy comparison, we take the median of the 50 paired attempt energy differences within each instance, then the median across instances, as defined in Supplementary Table 10 and main-text Eq. (24); the interval uses the same instance-then-attempt bootstrap with both medians recomputed in each resample. A two-sided paired sign test compares the directions of the instance-level medians, excluding ties within 10^{-4}. This needs no reference minimum, so all 40 quadratic instances enter, including instance 36. We also estimate success after repeated attempts on each instance and then average, using Eq. (23). The count n_{90} is the smallest integer n with $\bar{P}(n) \geq 0.9$. These curves insert observed success fractions in place of unknown probabilities and assume independent attempts with the same success probability on each instance. If a curve approaches a value below 0.9, the estimate has no threshold crossing. Finite data do not establish that the true restart count is infinite.

To check sensitivity to limited sampling, we replace each observed fraction by $p_i^{\mathrm{J}} = (k_i + \frac{1}{2})/(n_{\mathrm{att}} + 1)$, where k_i is the number of successes in $n_{\mathrm{att}} = 50$ attempts. This is Jeffreys smoothing: the posterior mean under a $\mathrm{Beta}(\frac{1}{2}, \frac{1}{2})$ prior. We insert that single mean into the same restart formula, without propagating uncertainty in the probability through the calculation. The curve is therefore neither a posterior predictive curve nor a confidence bound on the true restart count. Its quadratic crossing at $n = 75$ depends on this smoothing choice.

The pooled index in Eq. (22) permits comparison with the development results; it is not the confirmation statistic. As allowed by the protocol, we did not fit the optional mixed-effects logistic model because its required library was not installed.

Supplementary Note 7 Verification of the packaged compiler

The package tests compare `walsh-elimination` version 0.1.0 with complete enumeration and with the research implementation (Supplementary Table 11). These tests are separate from the benchmark runs. Reported errors are the largest absolute differences observed in double-precision floating-point arithmetic (binary64), not exact-arithmetic certificates.

Supplementary Table 11: **Checks of the released compiler against enumeration and research code.** Results are reported for version 0.1.0. Errors are maximum absolute energy or reconstruction differences on the stated test sets. Bitwise agreement means identical stored values for the tested configurations only. Zero observed error or zero violations does not establish a bound for every floating-point calculation.

Check	Test set	Result
Walsh reconstruction of $-\lvert P\rvert$, zero coefficient threshold	$d = 1, \ldots, 7$	Maximum error 2.5×10^{-14}
One-step conditional minimum	17 920 cases	Maximum error 1.8×10^{-14}
Reverse substitution attains the conditional minimum	Enumeration checks	Maximum error 0
Repeated elimination equals the pointwise partial minimum	60 runs	Maximum error 1.8×10^{-14}
Minimum energy and lifting of minimizers	40 runs	Maximum error 1.8×10^{-14}
Output interaction-order cap	40 runs	0 violations
`partial_eliminate` versus research implementation	200 configurations	0 bitwise differences
`partial_eliminate_kcap` versus research implementation	120 configurations	0 bitwise differences
Per-step reconstruction residual ϵ_a	30 runs	Maximum 1.2×10^{-14}
Input Hamiltonian remains unchanged	30 runs	0 violations

For the optional incident-term cap, the package counts terms in a separate index that is kept current. Old entries in the compatibility index therefore cannot inflate r_a and block an eligible elimination. The order in which coefficients are added remains unchanged. The higher-order research scheduler already updates its term index correctly, and the basic research scheduler does not use the stale index length to decide eligibility. Package and research code agree bit for bit on all 320 tested configurations, but this comparison does not cover every production benchmark.

The updated release checks report 75 passing tests when the research repository is present; without it, 62 tests pass and 13 research-comparison tests are skipped. The worked example also passes. Two added tests check that the stored degree safely bounds the actual neighbour count and that direct Walsh summation is implemented as described. These test and documentation changes leave the compiler's executable logic unchanged.

Supplementary Note 8 Compiler implementation and computational cost

This note describes how spins are selected, how numerical errors are checked, and how the compiler operates. It then bounds the cost of an implementation using a fast Walsh–Hadamard transform (FWHT). The benchmark code instead sums the Walsh formula directly; the two costs must be distinguished.

8.1 Ordering rules and eligibility checks

For quadratic inputs, `min_degree` chooses the eligible spin with the smallest degree in a maintained fill graph. This graph initially connects spins that share a term; each elimination removes one vertex and connects all its neighbours to one another. Edges remain even when their coefficients cancel, so the graph can overestimate the actual neighbour count. Applying d_{cap} to this graph can therefore exclude an otherwise acceptable spin, but cannot admit one whose actual count exceeds the cap. Before accepting a step, the compiler also checks the neighbours in the surviving terms. These checks affect the schedule, while every accepted step still uses the exact rule of Eq. (6).

For higher-order inputs, `min_degree` instead selects the eligible spin contained in the fewest terms, subject to $r_a \le r_{\mathrm{cap}}$ and at most 16 neighbours in the stored graph. The `min_fill` rule uses the same eligibility checks but selects the step that adds the fewest stored edges for quadratic inputs, or the fewest new terms after a trial expansion and combination with existing terms for higher-order inputs. Random ordering visits each spin once in a shuffled order. It skips ineligible spins and does not revisit them. None of these rules guarantees the smallest possible elimination width.

8.2 Walsh elimination compiler

Supplementary Algorithms S1 and S2 give one elimination step and the greedy selection loop. Main-text Section 2.2 defines the ordering rules and limits. The pseudocode assumes exact arithmetic. The implementation uses double-precision floating-point arithmetic (binary64) and removes coefficients of P_a with magnitude at most 10^{-15}. It retains Walsh coefficients and combined Hamiltonian terms only when their magnitudes exceed 10^{-12}. During recovery, it sets $s_a = +1$ when $|P_a| \le 10^{-12}$, then evaluates every recovered state in the original Hamiltonian.

To check the Walsh expansion, write the local polynomial after removal of small coefficients as P_a^{num}, set $F_a^{\mathrm{num}} = -|P_a^{\mathrm{num}}|$, and let $\widetilde{F}_a$ be the retained Walsh expansion. We compare their values on every neighbour configuration and record the largest discrepancy,

$$\epsilon_a = \max_{\boldsymbol{\sigma}} \left| F_a^{\mathrm{num}}(\boldsymbol{\sigma}) - \widetilde{F}_a(\boldsymbol{\sigma}) \right|. \tag{S29}$$

In the original certified settings, this discrepancy is at most 2.7×10^{-15} per step and 1.7×10^{-14} cumulatively per instance. For the confirmation test, the corresponding maxima are 1.8×10^{-15} and 7.1×10^{-14} (Supplementary Note 6.1), all well below the 10^{-4} success tolerance.

In exact arithmetic, the magnitude of each discarded Walsh coefficient is at most $\|F_a^{\mathrm{num}} - \widetilde{F}_a\|_\infty$. However, this check starts after small coefficients have been removed from P_a. It also excludes removal of small coefficients after combining terms and the tolerance used to choose a sign during recovery. The table checks and recovered-state energies therefore test numerical consistency, not exact zeros, a bound on all rounding errors, or preservation of every ground state in floating-point arithmetic. The package tests in Supplementary Note 7 use different instances and do not replace these benchmark checks.

Supplementary Algorithm S1 WALSHELIMINATESPIN: remove one spin and store its recovery rule.

Require: Hamiltonian $H = \{(S, c_S)\}$; spin a
Ensure: Reduced Hamiltonian H; reverse rule ρ_a

$P_a \leftarrow \sum_{S \ni a} c_S \prod_{i \in S \setminus \{a\}} s_i$
Delete every term with $a \in S$ from H
if $P_a \equiv 0$ **then**
    $\rho_a \leftarrow$ "s_a free" ▷ either sign is optimal
    return (H, ρ_a)
end if
$d_a \leftarrow |\mathcal{N}_a|$, the number of distinct spins in P_a
Evaluate $F_a = -|P_a|$ on the 2^{d_a} configurations of $\mathbf{s}_{\mathcal{N}_a}$ ▷ table: $\mathcal{O}(kr_a 2^{d_a})$
Compute $\widehat{F}_a(U)$ by the direct sums in Eq. (10) ▷ reference code: $\mathcal{O}((d_a+1)4^{d_a})$
for all $U \subseteq \mathcal{N}_a$ with $\widehat{F}_a(U) \neq 0$ **do**
    $c_U \leftarrow c_U + \widehat{F}_a(U)$ ▷ merge with existing terms; $U = \varnothing$ is the constant
end for
Store the current P_a in ρ_a with the rule $s_a = -\operatorname{sgn} P_a(\mathbf{s}_{\mathcal{N}_a})$ ▷ at $P_a = 0$: either sign
return (H, ρ_a)

Supplementary Algorithm S2 SCHEDULEDWALSHELIMINATION: select and remove eligible spins using a greedy order.

Require: H; target (f or spin count); active limits; ordering rule
Ensure: Reduced Hamiltonian H; reverse schedule $\mathcal{R}$ (applied in reverse order); attained fraction f_{att}; structural statistics Σ

$\mathcal{R} \leftarrow ()$; $\Sigma \leftarrow \varnothing$
$\mathcal{C} \leftarrow$ spins of H whose elimination satisfies every active limit ▷ r_{cap} and d_{cap}: direct checks; exact output limits: trial elimination
while the target has not been reached **and** $\mathcal{C} \neq \varnothing$ **do**
    $a \leftarrow$ minimizer of the ordering score over $\mathcal{C}$
    $(H, \rho_a) \leftarrow$ WALSHELIMINATESPIN(H, a)
    Append ρ_a to $\mathcal{R}$; record d_a, the induced order, the term count, and the coefficient range in Σ
    $\mathcal{C} \leftarrow$ spins of H whose elimination satisfies every active limit
end while
$f_{\mathrm{att}} \leftarrow |\mathcal{R}|/N$ ▷ $f_{\mathrm{att}} < f$ when no eligible spin remains
return $(H, \mathcal{R}, f_{\mathrm{att}}, \Sigma)$

8.3 Cost bound with an FWHT and local updates

Proposition S1 (Cost bound with an FWHT and local updates)**.** *Consider a prescribed sequence of m accepted eliminations. Suppose that, at every step, the eliminated spin has at most D distinct neighbours and at most $R_\star$ incident terms. Let $K \geq 1$ bound the interaction order of the input and every reduced Hamiltonian, including the final residual. Use unit-cost coefficient arithmetic and comparisons, and count each coefficient, spin index, and pointer as one storage cell. Assume a Hamiltonian dictionary keyed by the spin set S, with constant-time lookup, insertion, and deletion apart from key processing. Use live spin-to-term incidence lists with constant-time insertion and deletion per incidence through stored links. An in-place FWHT implementation that processes induced coefficients and their keys one at a time can execute the sequence in*

$$\mathcal{O}\big(m\,(KR_\star + D + 1)\,2^D\big) \tag{S30}$$

operations. If the input and every reduced Hamiltonian contain at most $L_{\max}$ nonzero terms, this implementation has peak storage

$$\mathcal{O}\big(N + KL_{\max} + mKR_\star + 2^D\big)\,, \tag{S31}$$

including the Hamiltonian, the stored reverse rules, and one Walsh table. For fixed D, $R_\star$, and K, the cost of executing the accepted sequence is linear in m. If dictionary operations have only expected amortized constant cost, the time bound has the same qualification. These are arithmetic-operation and storage-cell bounds, not bit-complexity bounds. They exclude initial index construction, candidate scans, ordering data structures and updates, trial transforms for output caps, and the residual solve.

Proof. Copying the incident terms into the reverse record, collecting their neighbours, and deleting their incidence entries costs $\mathcal{O}(Kr_a{+}1)$ per step. Neighbour configurations are generated one at a time, so evaluating $-|P_a|$ costs $\mathcal{O}((Kr_a + 1)2^{d_a})$ without storing all configurations. The in-place FWHT costs $\mathcal{O}(d_a 2^{d_a})$; normalization adds $\mathcal{O}(2^{d_a})$. Ordering the neighbour indices, building output keys, merging terms, and updating their incidence lists costs $\mathcal{O}((d_a + 1)2^{d_a})$, since each subset key and its incidence entries use at most d_a spin indices. Substituting $r_a \leq R_\star$ and $d_a \leq D$, then summing over m steps, gives Eq. (S30).

A term of order at most K uses $\mathcal{O}(K)$ cells for its coefficient, key, and incidence links. During a one-at-a-time merge, every live key is in the support before or after that elimination, so at most $2L_{\max}$ keys are stored. The Hamiltonian and incidence lists therefore use $\mathcal{O}(KL_{\max})$ cells. Active-spin and temporary index bookkeeping use $\mathcal{O}(N)$ cells. Each reverse record stores $\mathcal{O}(1 + Kr_a)$ cells; the record headers use $\mathcal{O}(m)$ cells, which is absorbed by $\mathcal{O}(N)$ because $m \leq N$. One Walsh table uses $\mathcal{O}(2^D)$ cells, and the current configuration and subset key use $\mathcal{O}(d_a)$ further cells, also bounded by $\mathcal{O}(N)$. Summing these contributions gives Eq. (S31). □

The benchmark code does not use an FWHT. It sums the Walsh formula directly and separately builds the spin-product vector for every subset. Computing the coefficients therefore requires $\mathcal{O}((d_a + 1)4^{d_a})$ operations and $\mathcal{O}((d_a + 1)2^{d_a})$ temporary storage. The code also copies and scans the current Hamiltonian at each step. Measured preprocessing times include all these operations. Proposition S1 bounds an achievable implementation, not the full runtime or storage of the benchmark code.

Supplementary Note 9 Cubic Max-Cut protocol

Software rule and sign convention. To compare cut values at $N = 40$–150, we use the standalone Python/PyTorch routine `exp6_aoc_dt.py`. It implements the AOC fixed-point update rule [20],

$$\mathbf{x}_{t+1} = \alpha(t)\mathbf{x}_t + \beta W \tanh(\mathbf{x}_t) + \gamma(\mathbf{x}_t - \mathbf{x}_{t-1}) + \mathbf{b}. \tag{S32}$$

Here $\mathbf{x}_t$ is a continuous state rather than an Ising spin configuration, and the matrix W is symmetric with zero diagonal. For both formulations, the energy to be minimized is $C - \mathbf{s}^{\mathsf{T}} W \mathbf{s}/2$, where C is a constant; thus $W_{ij} = -1$ on each original edge. The code stores the sign-reversed Hamiltonian $K = -H$ and eliminates spins by maximizing K. Its local replacement $+|P_a^K|$ and reverse rule $s_a = \operatorname{sgn} P_a^K$, where P_a^K is the local polynomial of K, are equivalent to Eqs. (6) and (7) for minimizing H. During recovery, the code chooses $s_a = +1$ when $|P_a^K| < 10^{-12}$. We apply no further matrix normalization, and both formulations have $\mathbf{b} = 0$.

The calculation uses double-precision floating-point arithmetic (binary64), $\beta = 0.9$, $\gamma = 0.1$, and $\alpha(t) = 0.5(1 - t/5000)$ for $t = 0, \ldots, 4999$. Each entry of $\mathbf{x}_0$ is drawn independently and uniformly from $[-1, 1)$, and $\mathbf{x}_{-1} = \mathbf{x}_0$. No recorded trajectory meets the stopping condition $\|\mathbf{x}_{t+1} - \mathbf{x}_t\|_2 < 10^{-5}$, so every run uses all 5000 iterations. We read out $s_i = \operatorname{sgn} x_{5000,i}$, mapping zero to $+1$, and perform no further search over the discrete spins. The reported cuts therefore come from a fixed number of iterations, not from converged fixed points. This software test does not measure physical AOC dynamics, hardware time, energy use, or optical noise. The 16-spin limit belongs only to the separate test of which graphs fit that budget; the larger software search tests exceed it.

Graphs and elimination orders. We draw simple random cubic graphs with `networkx.random_regular_graph`, repeating the draw until the graph is connected. The instance index i starts at zero, and the graph seeds are $10000 + 1000N + i$ for the fit test, $60000 + 1000N + i$ for the AOC-rule test, and $30000 + 1000N + i$ for the greedy test. Each seed initializes a Python `random.Random` generator that supplies the graph-generator seeds. The dependency versions used for the original runs were not recorded.

For the fit test, we generate 100 graphs at each even $N \in \{18, 20, \ldots, 40\}$ and measure

$$\widehat{Y}(N) = \frac{1}{100} \sum_{i=0}^{99} \mathbf{1}\{N_{\mathrm{red},i} \le N_{\mathrm{chip}}\}, \tag{S33}$$

where $N_{\mathrm{red},i}$ is the number of spins left in graph i. Thus, $\widehat{Y}(N)$ is the observed fraction of graphs that fit the budget. All orders remove only spins with at most three neighbours in the stored fill graph. Because zero-weight couplings remain in that graph, its degree can exceed, but not undercount, the number of nonzero neighbours. The minimum-degree order, called the baseline in this code, chooses the lowest-degree eligible spin and breaks ties by vertex label. It is not the unreduced search baseline. The random order chooses uniformly from the eligible spins at each step. The `min_fill` rule minimizes the number of new stored edges, while `min_promotes` minimizes the number of neighbours whose predicted degree after the step exceeds three, including those already above three. Both rules break ties by degree and then vertex label.

Restart comparisons and statistics. The AOC-rule test uses 20 graphs per size at $N \in \{40, 80, 150\}$, with 200 restarts per formulation at the same fixed settings, including $\beta = 0.9$. The graph seed also initializes the NumPy random generator for each batch of initial states.

A second condition reuses the same 200 initial states within each formulation and sets $\beta_{\mathrm{red}} = 0.9\rho(W_{\mathrm{orig}})/\rho(W_{\mathrm{red}})$, where $\rho(W) = \max_j |\lambda_j(W)|$ is the spectral radius, while β_{orig} remains 0.9. Every saved field for the original formulation is identical in the two conditions, so the original rows represent 200 distinct restarts, not 400. Main-text Figure 5b reports the fixed-β condition. Including the rescaled condition leaves the best cut unchanged on every graph. At $N = 150$, the paired t-test uses differences in the number of cut edges, whereas the plot shows percentage differences, $100(Q_{\mathrm{red}} - Q_{\mathrm{orig}})/Q_{\mathrm{orig}}$. The reported p-value is not adjusted for multiple comparisons.

The separate greedy test uses 20 graphs per size at $N \in \{150, 200\}$ and 200 restarts per formulation, with restart seeds $0, \ldots, 199$. Initial spins are drawn independently with equal probabilities for ± 1, and both formulations use the same seed for each restart. Each sweep visits spins in a new random order and sets each spin to the sign of its local field for maximizing $K = -H$, leaving the spin unchanged when the field magnitude is at most 10^{-12}. The solver stops after a sweep without a flip or after three times the current spin count in sweeps. At restart count R, we first find the best cut among the first R saved restarts on each graph, then average over graphs. The bands show standard deviations across graphs with divisor 20; panel b instead uses sample standard deviations with divisor 19. Neither is a confidence interval for the mean.

Both search tests use the minimum-degree elimination order, compare the formulations on the same graphs, and evaluate each recovered state in the original graph. The capacity, AOC-rule, and greedy tests use separate graph samples. The greedy test concerns the effect of the reduced Hamiltonian, not an effect specific to the AOC update rule.

Supplementary References

[1] Endre Boros and Peter L Hammer. Pseudo-Boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225, 2002.

[2] Georg Hahn and Hristo Djidjev. Reducing binary quadratic forms for more scalable quantum annealing. In *2017 IEEE International Conference on Rebooting Computing (ICRC)*, pages 1–8. IEEE, 2017.

[3] Tatsuhiko Shirai and Nozomu Togawa. Spin-variable reduction method for handling linear equality constraints in Ising machines. *IEEE Transactions on Computers*, 72(8):2151–2164, 2023.

[4] Hamed Karimi, Gili Rosenberg, and Helmut G Katzgraber. Effective optimization using sample persistence: A case study on quantum annealers and various Monte Carlo optimization methods. *Physical Review E*, 96(4):043312, 2017.

[5] Sergey Bravyi, Alexander Kliesch, Robert Koenig, and Eugene Tang. Obstacles to variational quantum optimization from symmetry protection. *Physical Review Letters*, 125(26):260505, 2020.

[6] Jernej Rudi Finžgar, Aron Kerschbaumer, Martin J. A. Schuetz, Christian B. Mendl, and Helmut G. Katzgraber. Quantum-informed recursive optimization algorithms. *PRX Quantum*, 5(2):020327, 2024.

[7] Ulrich Schollwöck. The density-matrix renormalization group. *Reviews of Modern Physics*, 77(1):259–315, 2005.

[8] Gil Refael, Stefan Kehrein, and Daniel S Fisher. Spin reduction transition in spin-$3/2$ random Heisenberg chains. *Physical Review B*, 66(6):060402, 2002.

[9] Yves Crama, Pierre Hansen, and Brigitte Jaumard. The basic algorithm for pseudo-Boolean programming revisited. *Discrete Applied Mathematics*, 29:171–185, 1990.

[10] Rina Dechter. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 113(1-2):41–85, 1999.

[11] Stefan Boettcher. Numerical results for ground states of spin glasses on Bethe lattices. *The European Physical Journal B*, 31(1):29–39, 2003.

[12] Stefan Boettcher and James Davidheiser. Reduction of dilute Ising spin glasses. *Physical Review B*, 77(21):214432, 2008.

[13] Stefan Boettcher. Physics of the Edwards–Anderson spin glass in dimensions $d = 3, \ldots, 8$ from heuristic ground state optimization. *Frontiers in Physics*, 12:1466987, 2024.

[14] Matthew Kowalsky, Tameem Albash, Itay Hen, and Daniel A Lidar. 3-regular three-XORSAT planted solutions benchmark of classical and quantum heuristic optimizers. *Quantum Science and Technology*, 7(2):025008, 2022.

[15] Tom Packebusch and Stephan Mertens. Low autocorrelation binary sequences. *Journal of Physics A: Mathematical and Theoretical*, 49(16):165001, 2016.

[16] James S. Cummins, Hayder Salman, and Natalia G. Berloff. Ising Hamiltonian minimization: Gain-based computing with manifold reduction of soft spins vs quantum annealing. *Phys. Rev. Res.*, 7(1):013150, Feb 2025.

[17] Kirill P Kalinin and Natalia G Berloff. Computational complexity continuum within Ising formulation of NP problems. *Communications Physics*, 5(1):20, 2022.

[18] Zhaokai Li, Nikesh S Dattani, Xi Chen, Xiaomei Liu, Hengyan Wang, Richard Tanburn, Hongwei Chen, Xinhua Peng, and Jiangfeng Du. High-fidelity adiabatic quantum computation using the intrinsic Hamiltonian of a spin system: Application to the experimental factorization of 291311. Preprint at `https://arxiv.org/abs/1706.08061`, 2017.

[19] Bao Yan, Ziqi Tan, Shijie Wei, Haocong Jiang, Weilong Wang, Hong Wang, Lan Luo, Qianheng Duan, Yiting Liu, Wenhao Shi, et al. Factoring integers with sublinear resources on a superconducting quantum processor. Preprint at `https://arxiv.org/abs/2212.12372`, 2022.

[20] Kirill P Kalinin, Jannes Gladrow, Jiaqi Chu, James H Clegg, Daniel Cletheroe, Douglas J Kelly, Babak Rahmani, Grace Brennan, Burcu Canakci, Fabian Falck, et al. Analog optical computer for AI inference and combinatorial optimization. *Nature*, 645(8080):354–361, 2025.