A CONTINUOUS ENERGY ISING SOLVER LEVERAGING DIFFERENCE-OF-CONVEX PROGRAMMING

Anonymous authors
Paper under double-blind review

ABSTRACT

Finding the ground state of the Ising model is a fundamental combinatorial optimization problem with applications across physics, materials science, machine learning, and operations research. The problem is NP-hard, and many practical solvers are based on heuristic annealing methods. While these methods are scalable, they can be sensitive to cooling schedules and parameter tuning, and many lack comparable convergence guarantees. On the other hand, continuous relaxations such as semidefinite programming offer rigorous guarantees but scale poorly with problem size.

We propose a continuous relaxation in which the discrete spin space $\{-1, 1\}^n$ is extended to $\mathbb{R}^n$. A separable quartic attractor replaces the hard constraint; its minimizers are exactly the vertices of a scaled hypercube $\{-\lambda, \lambda\}^n$. After solving the resulting continuous problem, a discrete solution is obtained via a componentwise sign mapping. The sum of the Ising energy and the attractor admits a difference-of-convex polynomial structure. This yields the **D**ifference-**o**f-**C**onvex **H**amiltonian (**DOCH**), which requires one matrix-vector multiplication and a componentwise cube-root update. Using the analytic nature of the Hamiltonian and the Łojasiewicz gradient inequality, we prove that DOCH converges to a critical point in this parameter regime and, under a nonzero-coordinate and Jacobian-stability condition, that the limit is a strict local minimum, enjoying linear convergence. For accelerated DOCH (ADOCH), we prove finite-window envelope descent and an $\mathcal{O}(N^{-1/2})$ best-iterate stationarity bound in the strict DC regime, together with full-sequence convergence when the look-back window is set to zero ($q = 0$) from some finite iteration onward.

We implement the proposed solvers on GPU platforms ranging from edge devices to high-performance computing clusters. Across MAX-CUT and spin-glass benchmarks, matched-budget comparisons with annealing, physics-inspired, and recent sampling-based solvers show favorable runtime-quality trade-offs from 10^3 to 10^8 spins.

1 INTRODUCTION

Combinatorial optimization arises across diverse domains, including social networks Yuan et al. (2018), finance Orús et al. (2019), cryptography Wang et al. (2020), scheduling Rieffel et al. (2015), electronic circuit design Barahona et al. (1988), and biosciences Kell (2012); Robert et al. (2021). These problems are challenging due to the exponentially large solution space, which renders exact methods such as exhaustive search, branch-and-bound, branch-and-cut, and dynamic programming impractical at scale Bellman (1957). This has motivated growing interest in alternative computing approaches.

Many combinatorial optimization problems can be reformulated as Ising problems, where one minimizes a quadratic energy function over binary spin variables. This includes problems such as MAX-CUT, maximum independent set, vertex cover, and the traveling salesman problem, which can be expressed or reduced to this form Lucas (2014). This viewpoint has led to the development of specialized algorithms and hardware, known as Ising machines, for computing ground states Tanahashi et al. (2019). Beyond optimization, Ising ground-state computation is also central to statistical physics, providing insight into phase transitions, magnetism, and critical phenomena Ising (1925); Sherrington & Kirkpatrick (1975).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Originally introduced by Lenz and Ising as a mathematical model for ferromagnetism Ising (1925), the Ising model has found broader applications in materials science, particularly in molecular and spin-glass studies Sherrington & Kirkpatrick (1975). In this model, the energy function is given by

$$\mathcal{E}(s_1, \ldots, s_n) = -\frac{1}{2} \sum_{i,j=1}^{n} J_{ij} s_i s_j - \sum_{i=1}^{n} h_i s_i, \tag{1}$$

where $s = (s_1, \ldots, s_n)$ is the spin vector with $s_i \in \{-1, +1\}$ representing up/down spins, $J_{ij} = J_{ji}$ is the coupling coefficient between spins s_i and s_j with $J_{ii} = 0$, and h_i is the external field acting on spin s_i. We use $s^* \in \{-1, +1\}^n$ to denote a minimizer of (1), referred to as the *ground state* of the Ising model.

Solving large Ising models by brute force is computationally intractable, with no known polynomial-time algorithm Barahona (1982). As a result, practical methods focus on efficiently approximating the ground state rather than computing it exactly. This has led to a wide range of algorithmic heuristics and specialized hardware architectures. We refer the reader to Mohseni et al. (2022); Moy et al. (2022); Lo et al. (2023); Maher et al. (2024) for comprehensive surveys of classical and recent advances.

The most widely used technique for solving Ising models is simulated annealing (SA) Kirkpatrick et al. (1983); Isakov et al. (2015); Goto et al. (2018); Bunyk et al. (2014), a method inspired by quantum annealing Kadowaki & Nishimori (1998); Santoro et al. (2002); Das & Chakrabarti (2008). SA uses Monte Carlo sampling guided by the Boltzmann distribution to search for the ground state. However, it relies on a heuristic cooling schedule to explore the energy landscape. Moreover, a gradual temperature reduction is used, which can result in slow convergence Yavorsky et al. (2019), making SA less suited for solving large-scale problems. Variants such as noisy mean field annealing (NMFA) King et al. (2018); Veszeli & Vattay (2022) and mean field annealing from a random state (MARS) Yavorsky et al. (2019) aim to speed up convergence. Their performance can depend on approximation quality and careful hyperparameter tuning.

State-of-the-art techniques based on Hamiltonian dynamics, such as the Simulated Bifurcation Machine (SBM) Puri et al. (2017); Goto (2019); Goto et al. (2021); Kanao & Goto (2022); Wang et al. (2023) and its ballistic variant (bSB) Goto et al. (2021), provide highly parallelizable dynamics and have demonstrated strong empirical performance on large-scale instances, but remain sensitive to parameter tuning and cooling schedules. Coherent Ising Machines (CIMs) Inagaki et al. (2016); Tiunov et al. (2019); Honjo et al. (2021), which leverage optical hardware, enable fast analog evolution and scalable parallel processing, but require complex, precisely controlled hardware and are affected by noise and decoherence Yamamoto et al. (2017); Pramanik et al. (2024). More recently, neural-network and accelerator-based solvers have emerged Zhang et al. (2018); Niazi et al. (2024); Jiang et al. (2024); Ichikawa & Arai (2025); Sun et al. (2023). Parallel quasi-quantum annealing (QQA) combines a continuous relaxation with gradient-based annealing, whereas iSCO uses discrete Langevin/MCMC sampling. These methods provide strong empirical baselines, but their performance depends on the annealing or sampling budget and on hyperparameter selection.

The Ising problem can also be relaxed into a continuous domain using convex Beck & Teboulle (2000) and semidefinite relaxations Luo et al. (2010), which admit convergent algorithms. A standard example is the Goemans–Williamson (GW) SDP Goemans & Williamson (1995). Its 0.878 approximation ratio applies to MAX-CUT with nonnegative edge weights and randomized hyperplane rounding; it does not extend to general signed Ising interactions or deterministic sign rounding. More general quadratic problems admit a $2/\pi$ guarantee under positive-semidefiniteness assumptions Nesterov (1998), but no constant-factor statement follows for an arbitrary indefinite interaction matrix. Semidefinite methods are also computationally expensive and do not scale well to large instances.

As industrial optimization problems grow in size, there is a need for scalable Ising solvers with transparent optimization behavior. To address this gap, we develop a solver with a low-cost fixed-point update and a convergence guarantee for its continuous Hamiltonian in a precisely stated parameter regime, without an annealing-type cooling schedule (see Appendix Table 1). The main contributions are summarized below.

1. **Continuous relaxation.** We relax the binary spin constraints and augment the Ising energy with a smooth quartic attractor whose minimizers form the scaled hypercube $\{-\lambda, \lambda\}^n$. The resulting

objective admits a difference-of-convex polynomial decomposition in the DC regime, enabling first-order iterations that require a single matrix-vector multiplication and a componentwise cube root.

2. **Convergence guarantees.** By exploiting the difference-of-convex structure and the Łojasiewicz gradient inequality Łojasiewicz (1963), we prove that DOCH converges to a critical point when $\mathbf{J} + \alpha\mathbf{I} \succ \mathbf{0}$ (Theorem 4.5), and under some mild assumptions it converges linearly to a strict local minima (Proposition 4.6). For ADOCH, Proposition 4.7 gives finite-window stationarity and a best-iterate rate, while Theorem 4.8 proves full convergence after a $q = 0$ monotone-window restart. A nonzero critical fixed point is a strict local minimizer under the condition in Proposition 4.6.
3. **Performance at scale.** The proposed solvers are GPU-parallelizable and can be implemented from edge devices to high-performance computing clusters. We have run matched-protocol experiments including optimized SA, QQA, iSCO, bSB, SimCIM, SIA, FEM, and GW-SDP where feasible, together with certified-optimum tests and uncertainty statistics. The scalability study reaches a matrix-free 10^7-spin dense stress test and a sparse 10^8-spin instance; these synthetic experiments demonstrate the computational scale and ground-state quality relative to the SOTA solvers.

Table 1: Qualitative comparison of representative Ising solvers. "Empirical" denotes solution-quality evidence rather than an approximation guarantee. Conditional SDP guarantees depend on the problem class and rounding rule.

Ising Solvers			Computational Complexity	Scalability/ Parallelization	Discrete Evidence	Convergence Guarantee	Use of Cooling	Parameter Search
Annealing	Markov Chain Monte Carlo (MCMC)	SA, MARS, NMFA iSCO	low	yes	empirical	no	yes	multiple
	Simulated Hamiltonian Dynamics	bSB, QQA	low	yes	empirical	no	yes	multiple
		CIM, SimCIM	low	yes	empirical	no	yes	multiple
		SIA	low	yes	empirical	no	yes	multiple
Continuous Optimization	Convex Programming, SDP		high	no	conditional	yes	no	—
	DOCH/ADOCH (present work)		**low**	**yes**	empirical	**yes**	**no**	one scalar

2 Continuous Relaxation

We develop a continuous reformulation of the Ising model by relaxing the binary spin vector to $\mathbb{R}^n$. The proofs of the results in this section are provided in Appendix C.

As a first step, we encode the coupling coefficients in an $(n \times n)$ symmetric matrix $\mathbf{J}$. Since any n-spin Ising model with external field can be equivalently reduced into a $(n+1)$-spin Ising model with zero external field (see Appendix B), it suffices to work with the following homogeneous energy function:

$$\mathcal{E}(\boldsymbol{s}) = -\frac{1}{2}\boldsymbol{s}^\top \mathbf{J}\boldsymbol{s}. \tag{2}$$

We work with this model in the rest of the paper.

The discrete domain $\{-1, 1\}^n$ can be identified with the vertices of the unit hypercube in $\mathbb{R}^n$. Exploiting the homogeneity of $\mathcal{E}$, we extend the domain to

$$\mathbb{D} = \big\{\{-\lambda, \lambda\}^n : \lambda > 0\big\},$$

and consider the optimization problem

$$\min_{\boldsymbol{x}\in\mathbb{D}} \mathcal{E}(\boldsymbol{x}). \tag{3}$$

It is not difficult to see that if $\boldsymbol{x}^*$ is a minimizer of (3), then $\boldsymbol{s}^* = \operatorname{sign}(\boldsymbol{x}^*)$ is a ground state of (1). Thus, (3) provides an equivalent formulation in the sense that minimizers in $\mathbb{D}$ map to ground states via the sign function.

Despite this equivalence, optimization over $\mathbb{D}$ remains difficult since $\mathbb{D}$ is nonconvex and disconnected. To address this, we replace the hard constraint $\boldsymbol{x} \in \mathbb{D}$ with an attraction mechanism that promotes solutions whose components concentrate near a scaled hypercube $\{-\lambda, +\lambda\}^n$ in the continuous

domain. Specifically, for $\alpha, \beta > 0$, we introduce the quartic polynomial

$$\mathcal{A}(\boldsymbol{x}) = \frac{\beta}{4}\big(x_1^4 + \cdots + x_n^4\big) - \frac{\alpha}{2}\big(x_1^2 + \cdots + x_n^2\big), \tag{4}$$

where $\alpha, \beta > 0$ control the shape and scale of the attractor. Notice that $\mathcal{A}$ is not a nonnegative penalty and does not vanish on the binary hypercube. Its role is instead determined by its minimizer set, characterized below. Appendix G gives a matrix-dependent rule for β and an effective selection procedure for α.

The specific form of the attractor (4) is motivated by the Hamiltonian of nonlinear parametric oscillators Goto (2019) (see Appendix D). The following proposition characterizes the global minimizers of the attractor and links it directly to the scaled domain $\mathbb{D}$.

Proposition 2.1. *The set of global minimizers of $\mathcal{A}$ is $\{-\lambda, +\lambda\}^n$, where $\lambda = \sqrt{\alpha/\beta}$.*

In particular, the minimizers of $\mathcal{A}$ coincide with the scaled hypercube $\mathbb{D}$. The scale $\lambda = \sqrt{\alpha/\beta}$, and hence the location of these minimizers, is determined by the parameters α and β, which control the target set induced by the attractor. As a result, the attractor promotes solutions close to $\mathbb{D}$ and provides a soft replacement for the hard constraint $\boldsymbol{x} \in \mathbb{D}$.

We combine the attractor $\mathcal{A}$ into the Ising energy and define the *Hamiltonian* $\mathcal{H} = \mathcal{A} + \mathcal{E}$. Specifically, we consider the following unconstrained optimization problem

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \mathcal{H}(\boldsymbol{x}) = \mathcal{A}(\boldsymbol{x}) + \mathcal{E}(\boldsymbol{x}). \tag{5}$$

Unlike the original Ising problem (2), which has a finite search space, the continuous optimization problem (5) may not have a minimizer. However, the following result shows that a global minimizer always exists.

Proposition 2.2. *The Hamiltonian $\mathcal{H}$ is coercive and bounded below on $\mathbb{R}^n$, and therefore admits a global minimizer.*

This establishes that the relaxed problem is well-defined. The next result connects minimizers of $\mathcal{H}$ back to solutions of the original Ising problem when they align with the scaled hypercube.

Proposition 2.3. *Suppose $\boldsymbol{x}^*$ is a minimizer of $\mathcal{H}$ and lies on a vertex of the scaled hypercube $\{-\lambda, \lambda\}^n$, where $\lambda > 0$. Then $\operatorname{sign}(\boldsymbol{x}^*)$ is a ground state of the Ising model* (2)*.*

Together, these results imply that the relaxed formulation recovers the exact Ising solution whenever a global minimizer of $\mathcal{H}$ lies on $\mathbb{D}$. In general, neither a minimizer nor a critical point of $\mathcal{H}$ need lie on $\mathbb{D}$. Consequently, deterministic sign rounding can return a suboptimal spin vector, and we do not claim a general bound relating its Ising energy to the ground-state energy. The parameters α and β steer continuous solutions toward $\mathbb{D}$; the quality of the rounded solution is evaluated empirically in Section 5 and Appendix M.

3 First-Order Algorithms

Since the Hamiltonian $\mathcal{H}$ is smooth, a natural approach is to optimize it using first-order methods Beck (2017); Nesterov (2018). This observation motivates the continuous relaxation.

First-order methods are widely used in large-scale optimization due to their low per-iteration cost and scalability to high-dimensional problems LeCun et al. (2015); Beck (2017). They rely only on gradient information, avoiding Hessian computations and enabling efficient implementation in large-scale settings.

However, direct application of these methods is not straightforward in our setting. For the polynomial Hamiltonian,

$$\nabla^2 \mathcal{H}(\boldsymbol{x}) = 3\beta \operatorname{diag}(x_1^2, \ldots, x_n^2) - (\mathbf{J} + \alpha \mathbf{I}),$$

the Hessian norm grows unboundedly as $\|\boldsymbol{x}\| \to \infty$, so $\nabla \mathcal{H}$ is not globally Lipschitz. As a consequence, standard step-size guarantees for gradient descent do not apply, and stable step-size selection becomes nontrivial. Adaptive methods introduce additional hyperparameters but still lack convergence guarantees in this regime. Second-order methods are also impractical at scale due to the cost of solving a large linear system at each iteration.

To overcome these limitations, we exploit the structure of $\mathcal{H}$ by rewriting it as a difference of convex functions. More precisely, combining the contributions of $\mathcal{A}$ and $\mathcal{E}$, we obtain

$$\mathcal{H}(\boldsymbol{x}) = \frac{\beta}{4}\sum_{i=1}^{n} x_i^4 - \frac{\alpha}{2}\|\boldsymbol{x}\|^2 - \frac{1}{2}\boldsymbol{x}^\top \mathbf{J}\boldsymbol{x} = f(\boldsymbol{x}) - g(\boldsymbol{x}),$$

where

$$f(\boldsymbol{x}) = \frac{\beta}{4}\sum_{i=1}^{n} x_i^4, \qquad g(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top(\mathbf{J} + \alpha\mathbf{I})\boldsymbol{x}.$$

Proposition 3.1. *The function f is convex for all $\beta > 0$. Let $\alpha_0 = \lambda_{\max}(-\mathbf{J}) = -\lambda_{\min}(\mathbf{J}) \geqslant 0$. Then g is convex for $\alpha \geqslant \alpha_0$ and strongly convex for $\alpha > \alpha_0$.*

When $\alpha \geqslant \alpha_0$, the problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \; f(\boldsymbol{x}) - g(\boldsymbol{x}) \tag{6}$$

falls under difference-of-convex programming (DCP) Abbaszadehpeivasti et al. (2023); Phan et al. (2018). Strong convexity, and hence the convergence analysis in Section 4, requires the strict inequality $\alpha > \alpha_0$. Outside this DC regime the same fixed-point map remains well-defined, but the descent theorem does not apply. Empirically, it also achieves a better runtime-quality trade-off compared to GD and ADAM (see Figure 20).

3.1 DOCH Algorithm

Our Ising solver is based on the DCP formulation (6). A standard approach for such problems is the Difference-of-Convex Algorithm (DCA) An & Tao (2005); Abbaszadehpeivasti et al. (2023), which can be interpreted as an instance of the majorization-minimization principle Dempster et al. (1977). At each iteration, DCA constructs a convex surrogate that upper-bounds the objective and is easier to minimize.

Applying this construction to (6) yields the iterative scheme in Algorithm 1 (see Appendix E for derivation), with update map

$$\mathcal{T}(\boldsymbol{x}) = \varphi\big(\beta^{-1}(\mathbf{J} + \alpha\mathbf{I})\boldsymbol{x}\big), \qquad \varphi(x_1, \ldots, x_n) = (\sqrt[3]{x_1}, \ldots, \sqrt[3]{x_n}).$$

Thus, each iteration consists of a linear transformation followed by a componentwise cube-root nonlinearity. We refer to the resulting method as the **Difference-of-Convex Hamiltonian** (DOCH) algorithm.

Algorithm 1 DOCH

Input: initial point $\boldsymbol{x}^{(0)}$, number of iterations $N \geqslant 1$
for $k = 0$ **to** $N - 1$ **do**
  Compute $\boldsymbol{x}^{(k+1)} = \mathcal{T}(\boldsymbol{x}^{(k)})$
end for
Output: $\boldsymbol{s} = \operatorname{sign}(\boldsymbol{x}^{(N)})$

A two-spin example illustrating the DOCH update steps is shown in Figure 2, where we use a two-spin Ising model to visualize the updates and the convergence of the iterates.

In the strict DC regime $\mathbf{J} + \alpha\mathbf{I} \succ \mathbf{0}$, a distinctive property of DOCH is that the updates are monotone:

$$\mathcal{H}(\boldsymbol{x}^{(0)}) \geqslant \mathcal{H}(\boldsymbol{x}^{(1)}) \geqslant \mathcal{H}(\boldsymbol{x}^{(2)}) \geqslant \cdots. \tag{7}$$

That is, $\mathcal{H}$ is guaranteed to decrease or stay constant after each iteration. Since $\mathcal{H}$ is continuous and coercive (Proposition 2.2), the descent property (7) ensures that the sequence $\{\mathcal{H}(\boldsymbol{x}^{(k)})\}$ converges to a limiting value. Moreover, because $\mathcal{H}$ is a polynomial, we can establish that the sequence of iterates $\{\boldsymbol{x}^{(k)}\}$ converges to a limit point $\boldsymbol{x}^* \in \mathbb{R}^n$.

In Section 4, we show that $\boldsymbol{x}^*$ is a critical point of $\mathcal{H}$. Proposition 4.6 gives additional, explicitly checkable conditions for a critical fixed point to be a strict local minimizer.

3.2 ADOCH ALGORITHM

A natural extension of our algorithm is to incorporate acceleration into DCA Phan et al. (2018). Accelerated first-order methods can improve the empirical convergence speed of standard schemes while preserving their low per-iteration cost. A prominent example is Nesterov's acceleration Nesterov (2018), which achieves the optimal rate for a class of convex problems.

In our setting, we apply a Nesterov-style acceleration to DCA, leading to the Accelerated DOCH (ADOCH) algorithm, with the main steps summarized in Algorithm 2.

Algorithm 2 Accelerated DOCH (ADOCH)

Input: $\boldsymbol{x}^{(0)}$, $\boldsymbol{y}^{(0)} = \boldsymbol{x}^{(0)}$, $t_0 = 1$, window size $q \geqslant 0$, iterations $N \geqslant 1$
for $k = 0$ **to** $N - 1$ **do**
  Compute $t_{k+1} = \big(1 + \sqrt{1 + 4t_k^2}\big)/2$
  if $k \geqslant 1$ **then**
    Compute $\boldsymbol{y}^{(k)} = \boldsymbol{x}^{(k)} + \dfrac{t_k - 1}{t_{k+1}} \big(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\big)$
  end if
  if $\mathcal{H}(\boldsymbol{y}^{(k)}) \leqslant \max\big\{\mathcal{H}(\boldsymbol{x}^{(\max(0,k-q))}), \ldots, \mathcal{H}(\boldsymbol{x}^{(k)})\big\}$ **then**
    Set $\boldsymbol{v}^{(k)} = \boldsymbol{y}^{(k)}$
  else
    Set $\boldsymbol{v}^{(k)} = \boldsymbol{x}^{(k)}$
  end if
  Compute $\boldsymbol{x}^{(k+1)} = \mathcal{T}(\boldsymbol{v}^{(k)})$
end for
Output: $\boldsymbol{s} = \operatorname{sign}(\boldsymbol{x}^{(N)})$

The main difference with DOCH is the use of momentum-based extrapolation. Specifically, at iteration k, we first form an extrapolated point

$$\boldsymbol{y}^{(k)} = \boldsymbol{x}^{(k)} + \frac{t_k - 1}{t_{k+1}} \big(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\big),$$

where the sequence $\{t_k\}$ is set using Nesterov's optimal scheme Nesterov (2018). We then decide whether to accept $\boldsymbol{y}^{(k)}$ using a look-back criterion:

$$\mathcal{H}(\boldsymbol{y}^{(k)}) \leqslant \max\big\{\mathcal{H}(\boldsymbol{x}^{(\max(0,k-q))}), \ldots, \mathcal{H}(\boldsymbol{x}^{(k)})\big\}, \tag{8}$$

where q controls the window size Phan et al. (2018). The choice $q = 0$ gives a monotone window, whereas $q > 0$ permits non-monotone accepted values. The Hamiltonian in this test is evaluated exactly. The product $\mathbf{J}\boldsymbol{y}^{(k)}$ is cached and reused if the extrapolated point is accepted; if it is rejected, the already available product at $\boldsymbol{x}^{(k)}$ is reused. Thus the acceptance test does not require a second matrix-vector multiplication for the update. If the condition holds, we set $\boldsymbol{x}^{(k+1)} = \mathcal{T}(\boldsymbol{y}^{(k)})$; otherwise, we use $\boldsymbol{x}^{(k+1)} = \mathcal{T}(\boldsymbol{x}^{(k)})$. This adaptive-restart rule is motivated by non-monotone acceleration schemes Grippo & Sciandrone (2002); Phan et al. (2018).

4 CONVERGENCE ANALYSIS

In this section, we present a convergence analysis of DOCH. The analysis builds on standard properties of difference-of-convex optimization, adapted to the specific polynomial structure of $\mathcal{H}$. The proofs of results in this section are provided in Appendix F.

We first establish a basic descent property, which is the starting point of the analysis. Throughout this section we assume the strict DC condition $\alpha > \lambda_{\max}(-\mathbf{J})$, so that g is strongly convex, and define

$$\mu = \lambda_{\min}(\mathbf{J} + \alpha\mathbf{I}) > 0. \tag{9}$$

Proposition 4.1. *Let $\{\boldsymbol{x}^{(k)}\}$ be the DOCH iterates. Then*

$$\mathcal{H}(\boldsymbol{x}^{(k)}) - \mathcal{H}(\boldsymbol{x}^{(k+1)}) \geqslant \frac{\mu}{2}\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|^2. \tag{10}$$

In particular, this yields the descent property (7), which ensures that the objective values decrease along the iterates. However, this alone does not control the behavior of the iterates themselves. For example, it does not rule out the divergence of the iterates. A key step is therefore to establish boundedness. In standard DCA analyses, boundedness is assumed Phan et al. (2018). For DOCH, boundedness can be established from the coercivity of the Hamiltonian.

Proposition 4.2. *The DOCH iterates are bounded, and*

$$\lim_{k\to\infty} \|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_2 = 0. \tag{11}$$

The next result shows that if the increments of the iterates vanish, then the gradient norm also vanishes, which is a necessary condition for convergence to a critical point.

Proposition 4.3. *There exists $\sigma > 0$ such that for all $k \geqslant 1$,*

$$\|\nabla\mathcal{H}(\boldsymbol{x}^{(k)})\|_2 \leqslant \sigma\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\|_2. \tag{12}$$

Combined with boundedness, this implies that any limit point of $\{\boldsymbol{x}^{(k)}\}$ is a critical point of $\mathcal{H}$.

Proposition 4.4. *All limit points of $\{\boldsymbol{x}^{(k)}\}$ are critical points of $\mathcal{H}$.*

At this stage, the analysis reaches a natural stopping point within the standard descent framework: while it guarantees asymptotic stationarity, it does not ensure convergence of the full sequence. To proceed further, additional structure on $\mathcal{H}$ is required beyond smoothness and descent. In particular, one needs an analytic property that links the geometry of $\mathcal{H}$ to its gradient behavior near critical points.

This is provided by the Łojasiewicz gradient inequality, which is central to upgrading subsequential convergence to full convergence. Specifically, since $\mathcal{H}$ is a polynomial and hence analytic, it satisfies the Łojasiewicz gradient inequality Łojasiewicz (1963). This enables us to strengthen the above subsequential convergence result to convergence of the full sequence Absil et al. (2005); Attouch et al. (2013).

Theorem 4.5. *Suppose $\alpha > \lambda_{\max}(-\mathbf{J})$ and $\beta > 0$. Then the DOCH iterates $\{\boldsymbol{x}^{(k)}\}$ converge to a critical point of $\mathcal{H}$.*

The result can be strengthened into linear convergence to a strict local minima under additional assumptions.

Proposition 4.6. *Let $\boldsymbol{x}^*$ be a critical fixed point of $\mathcal{T}$. Suppose tha $\|\nabla\mathcal{T}(\boldsymbol{x}^*)\|_2 < 1$, and assume that all components of $\boldsymbol{x}^*$ are nonzero. Then $\boldsymbol{x}^*$ is a strict local minimizer of $\mathcal{H}$, and $\mathcal{H}$ satisfies the Kurdyka–Łojasiewicz inequality at $\boldsymbol{x}^*$ with exponent $\theta = 1/2$. Consequently, in the strict DC regime, any DOCH sequence converging to $\boldsymbol{x}^*$ does so linearly.*

Overall, DOCH is a provably convergent first-order method for approaching a critical point of the nonconvex Hamiltonian $\mathcal{H}$ in the strict DC regime, despite the absence of global Lipschitz smoothness. Proposition 4.6 is a local property of any critical fixed point and can also be checked outside that regime; it does not by itself prove that the iterates converge there.

We next give guarantees for ADOCH. Set $\mathbf{A} = \mathbf{J} + \alpha\mathbf{I}$, and for Algorithm 2 define the accepted-point residual and window envelope

$$\boldsymbol{d}_k = \boldsymbol{x}^{(k+1)} - \boldsymbol{v}^{(k)}, \qquad M_k = \max_{\max(0,k-q)\leqslant i\leqslant k} \mathcal{H}(\boldsymbol{x}^{(i)}). \tag{13}$$

Unlike $\mathcal{H}(\boldsymbol{x}^{(k)})$, the envelope M_k is a Lyapunov sequence for every fixed finite q.

Proposition 4.7. *Suppose $\mathbf{A} \succeq \mathbf{0}$, $\beta > 0$, and $q < \infty$. Let $\mu = \lambda_{\min}(\mathbf{A}) \geqslant 0$. Then the ADOCH iterates are bounded, $\{M_k\}$ is non-increasing, and*

$$\mathcal{H}(\boldsymbol{x}^{(k+1)}) \leqslant M_k - \frac{\mu}{2}\|\boldsymbol{d}_k\|_2^2 - \frac{\beta}{12n}\|\boldsymbol{d}_k\|_2^4. \tag{14}$$

Moreover,

$$\lim_{k\to\infty} \min_{k\leqslant j\leqslant k+q} \|\nabla\mathcal{H}(\boldsymbol{x}^{(j+1)})\|_2 = 0. \tag{15}$$

If $\mathbf{A} \succ \mathbf{0}$*, then, with* $\mathcal{H}_{\inf} = \inf_{\boldsymbol{x}\in\mathbb{R}^n} \mathcal{H}(\boldsymbol{x})$*, for every* $N \geqslant 0$*,*

$$\min_{0\leqslant j\leqslant N+q} \|\nabla\mathcal{H}(\boldsymbol{x}^{(j+1)})\|_2 \leqslant \|\mathbf{A}\|_2 \sqrt{\frac{2(q+1)(\mathcal{H}(\boldsymbol{x}^{(0)}) - \mathcal{H}_{\inf})}{\mu(N+1)}}. \tag{16}$$

Thus the non-monotone experimental choice $q > 0$ has a finite-window stationarity certificate and an $\mathcal{O}(N^{-1/2})$ best-iterate rate, but Proposition 4.7 alone does not assert convergence of every iterate. Full convergence is recovered by the monotone-window mode.

Theorem 4.8. *Suppose* $\mathbf{A} \succeq \mathbf{0}$ *and* $\beta > 0$*. If ADOCH uses* $q = 0$ *from the beginning, or switches permanently to* $q = 0$ *after any finite iteration, then* $\{\boldsymbol{x}^{(k)}\}$ *converges to a critical point of* $\mathcal{H}$*.*

The residual $\boldsymbol{d}_k$, rather than $\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$, is the corresponding stationarity certificate. Detailed proofs, the semidefinite $\mathcal{O}(N^{-1/4})$ bound and unconditional boundedness outside the DC regime are given in Appendix F.

5 Experiments

We evaluate DOCH and ADOCH on Max-Cut and spin-glass instances across five NVIDIA GPUs (Jetson Nano through H100) against optimized SA Isakov et al. (2015), QQA Ichikawa & Arai (2025), iSCO Sun et al. (2023), GW-SDP Goemans & Williamson (1995), bSB Goto et al. (2021), SimCIM Inagaki et al. (2016); Goto et al. (2021), SIA Jiang et al. (2024), and FEM Shen et al. (2025). Solvers use a matched adaptive-search budget and, unless stated otherwise, 100 initializations. Exact ADOCH acceptance and final-energy evaluation are included. Tuning and other implementation details are given in Appendices I, J, and M.

Small-Scale Models. On the 10^3-spin Sherrington–Kirkpatrick (SK) model Sherrington & Kirkpatrick (1975), DOCH and ADOCH reach the reference energy from 100 GW-SDP projections in $\approx$2.5 and 3.5 ms, respectively (Appendix Figure 5, left). This diagnostic threshold is not a certificate of optimality.

On K_{2000}, DOCH and ADOCH reach the GW-SDP reference in $\approx$1.9 and 2.6 ms (Appendix Figure 5, right). Because the displayed SA trace is short, the final comparison instead uses optimized SA Isakov et al. (2015) under the same wall-clock budget; DOCH obtains the better objective on all five tested 10^3–2×10^3 instances (Appendix Table 6).

We use the equivalent Ising formulation of MAX-CUT (Appendix H) on the G_{10} graph (800 nodes, 5.99% connectivity, ± 1 weights). DOCH and ADOCH reach the best observed cut within $\approx$100 iterations; ADOCH reaches the GW-SDP reference level after three iterations on average (Figure 6a). Additional G-set results with max–min intervals and normalization by published cut values are given in Figure 18.

Figure 19 reports a diagnostic average time to reach 99% of the best cut found by 10^5 SA iterations on random integer-weighted Biq Mac graphs Wiegele (2007). Because that target is solver-defined, we do not interpret it as a certified TTS. We additionally evaluate single-run TTS to certified global optima on small Rudy/Biq Mac, K_{22}, and SK_{24} instances. DOCH reaches the exact optimum in every reported run and is faster than optimized SA on all four tests (Table 7).

Medium-to-Large Models. For $n > 10^4$, GW-SDP is computationally infeasible, and we use RTX 3090 or V100 GPUs. On the 10^4-spin SK model (Figure 6b), ADOCH achieves the lowest energy overall, while DOCH reaches energy below -5×10^5 in under 0.1 s. Results for additional 10^5 and 10^6 spins Ising models are given in Figures 10–15 of Appendix N.

Against the official FEM implementation Shen et al. (2025), DOCH and ADOCH obtain lower energies on the tested 10^3- and 10^4-spin instances within 0.1 s and 1 s budgets (Figures 7 and 8). Matched comparisons with QQA and multi-chain iSCO are reported in Tables 4 and 5; DOCH has the best normalized objective on the four reported Complete K_n graphs and SK models, with ADOCH close to the best.

Ultra-Large Models. On a matrix-free 10^7-spin model with $J_{ij} = \sin(ij + \text{seed})$ and $\approx$50 trillion implicit couplings, ADOCH obtains the lowest energy among the compared solvers, followed by DOCH, on four H100 GPUs (Figure 6c). This 11-hour procedural stress test is not representative

Table 2: **Summary of numerical results across solver families and problem scales.** For objective/cut ratios, ↑ is better; for Ising energy, runtime, and time-to-solution (TTS), ↓ is better. Ratios in the QQA/iSCO and fixed-budget experiments are normalized within the corresponding comparison and are *not* approximation guarantees. Exact tabulated values are used where available; ≈ denotes values read from the Appendix curves.

Benchmark / setting	Metric	Compared baselines	DOCH	ADOCH
Small-scale references and certified tests (Figure 5; Tables 6, 7, 8)				
SK_{1000} / K_{2000}	Time to reach best GW-SDP reference ↓	GW-SDP reference from 100 randomized projections	**2.5 / 1.9 ms**	3.5 / 2.6 ms
5 instances, $n = 10^3$–2×10^3	Equal-wall-clock objective ratio ↑	Optimized SA: 0.966–0.987	**1.000 on 5/5**	–
4 certified instances, $n \leqslant 24$	TTS to exact global optimum (ms) ↓	SA: 0.52–4.03	**0.04–0.10**	0.07–0.14
Planted signed-bipartite, $n \leqslant 1000$	Best ratio ↑; mean $\mathcal{E}$ at $n = 1000$	Certified $\mathcal{E}^\star = -\mathbf{250{,}000}$	**1.000**; $-247{,}499.84$	**1.000**; $-5{,}000.00$
Matched modern baselines (Tables 4, 5; Figure 7)				
$K_{10\,000}$ MAX-CUT	Mean objective ratio ↑ / runtime (s)	QQA: 0.924/6.48; iSCO: 0.892/1.13	**1.000**/1.99	0.994/1.17
$SK_{10\,000}$	Mean objective ratio ↑ / runtime (s)	QQA: 0.924/6.66; iSCO: 0.884/1.12	**1.000**/2.00	0.993/1.16
SK_{1000}, $t = 10$ ms	Ising energy ($\times10^4$) ↓	SA ≈ −0.06, bSB ≈ −1.42, SimCIM ≈ −1.33, SIA ≈ −1.30, FEM ≈ −1.40	≈ **-1.58**	≈ −1.55
Medium/large-scale robustness and scalability (Figures 16, 17, 18, 20)				
5 models, $n = 10^4$–10^5	Mean fixed-budget energy-ratio range ↑	bSB ≈ 0.86–0.94; SimCIM ≈ 0.88–0.93; SIA ≈ 0.85–0.92	≈ 0.90–0.95	≈ **0.94–0.97**
G-set, 10 graphs, 100 iterations	Mean cut-ratio range ↑	bSB ≈ 0.86–0.95; SimCIM ≈ 0.91–0.94; SIA ≈ 0.83–0.92	≈ 0.90–0.94	≈ 0.90–0.94
10^7 spins, 0.001% connectivity	Last plotted $\mathcal{E}$ ($\times10^{10}$) ↓	SA ≈ 0, bSB ≈ −2.1, SimCIM ≈ −1.55, SIA ≈ −2.65	≈ −2.65	≈ **−2.90**
K_{2000} optimization baseline	Parameter-search time (s) ↓	Vanilla GD ≈ 1.3×10^3; Adam ≈ 1.6×10^3	≈ **5**	≈ **5**

of an arbitrary dense matrix, and its Wigner-based parameter is uncertified. On a stored 10^8-spin sparse model (≈0.5B nonzeros), both methods obtain the lowest compared energy within 10^3 s on two H100 GPUs (Figure 6d). Neither instance has a certified optimum or independent bound; details and further results are given in Figure 16.

6 Conclusion

We introduced DOCH and ADOCH, continuous Ising solvers that combine the Ising energy with a quartic attractor. In the strict DC regime, DOCH converges to a critical point of the relaxed Hamiltonian, and, under an additional stability condition, converges linearly to a strict local minimum. ADOCH satisfies finite-window envelope descent and stationarity guarantees; with a permanent $q = 0$ monotone-window restart, its full iterate sequence converges. For a fixed non-monotone window $q > 0$, Proposition 4.7 guarantees only windowed stationarity and a best-iterate rate, while Theorem 4.8 requires the permanent $q = 0$ switch for full-sequence convergence. Proposition 4.6 further requires nonzero coordinates and a Jacobian spectral-norm condition to identify a strict local minimum. All of these guarantees apply to the continuous relaxed Hamiltonian, not to the discrete sign-rounded objective: in particular, they do not bound the energy of $\mathrm{sign}(\boldsymbol{x}^*)$, for which a general deterministic-rounding approximation guarantee is difficult to obtain (Table 1).

Both methods require only one matrix-vector product per update, plus componentwise operations, and they parallelize across compact GPUs and multi-GPU systems for both stored-sparse and matrix-free interactions. Annealing methods may remain attractive on small instances, where many sweeps and careful cooling schedules are affordable. DOCH and ADOCH instead emphasize rapid, cooling-free optimization, with their main advantage emerging on large and ultra-large Ising models where prolonged annealing becomes costly. Thus, they complement established annealing approaches while extending efficient Ising optimization to substantially larger regimes.

AI USE STATEMENT

In this work, we used generative AI tools only for textual editing: polishing phrasing, improving clarity and flow, and correcting grammar and LaTeX formatting. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We describe all the methods, experimental setup, training, datasets, and evaluation protocols in the main paper and appendix. Complete proofs are provided in the appendix. The source code for DOCH and ADOCH solvers are included as supplementary material and will be released publicly upon acceptance.

REFERENCES

H. Abbaszadehpeivasti, E. de Klerk, and M. Zamani. On the rate of convergence of the difference-of-convex algorithm (DCA). *Journal of Optimization Theory and Applications*, 2023.

Pierre-Antoine Absil, Robert Mahony, and Ben Andrews. Convergence of the iterates of descent methods for analytic cost functions. *SIAM Journal on Optimization*, 16(2):531–547, 2005.

Le Thi Hoai An and Pham Dinh Tao. The dc (difference of convex functions) programming and DCA revisited with dc models of real world nonconvex optimization problems. *Annals of Operations Research*, 133:23–46, 2005.

Hedy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. *Mathematical Programming*, 137(1):91–129, 2013.

Francisco Barahona. On the computational complexity of Ising spin glass models. *Journal of Physics A: Mathematical and General*, 15(10):3241–3253, 1982.

Francisco Barahona, Martin Grötschel, Michael Jünger, and Gerhard Reinelt. An application of combinatorial optimization to statistical physics and circuit layout design. *Operations Research*, 36(3):493–513, 1988.

Amir Beck. *First-order methods in optimization*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, October 2017. ISBN 978-1-61197-498-0.

Amir Beck and Marc Teboulle. Global optimality conditions for quadratic optimization problems with binary constraints. *SIAM Journal on Optimization*, 11(1):179–188, 2000.

Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957.

Paul I. Bunyk, Emile M. Hoskinson, Mark W. Johnson, Elena Tolkacheva, Fabio Altomare, and Andrew J. Berkley. Architectural considerations in the design of a superconducting quantum annealing processor. *IEEE Transactions on Applied Superconductivity*, 24(4):1–10, 2014.

David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Undergraduate Texts in Mathematics. Springer, Cham, 4 edition, 2015.

Arnab Das and Bikas K. Chakrabarti. Quantum annealing and analog quantum computation. *Reviews of Modern Physics*, 80:1061–1081, Sep 2008.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–22, 1977.

Pierre Frankel, Guillaume Garrigos, and Juan Peypouquet. Splitting methods with variable metric for Kurdyka–Łojasiewicz functions and general convergence rates. *Journal of Optimization Theory and Applications*, 165(3):874–900, 2015.

Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

H. Goto, K. Endo, M. Suzuki, Y. Sakai, T. Kanao, Y. Hamakawa, R. Hidaka, M. Yamasaki, and K. Tatsumura. High-performance combinatorial optimization based on classical mechanics. *Science Advances*, 7:eabe7953, 2021.

Hayato Goto. Quantum computation based on quantum adiabatic bifurcations of kerr-nonlinear parametric oscillators. *Journal of the Physical Society of Japan*, 2018.

Hayato Goto. Quantum computation based on quantum adiabatic bifurcations of Kerr-nonlinear parametric oscillators. *Journal of the Physical Society of Japan*, 88(6):061015, 2019.

Hayato Goto, Zhirong Lin, and Yasunobu Nakamura. Boltzmann sampling from the ising model using quantum heating of coupled nonlinear oscillators. *Scientific Reports*, 8(1):7154, 2018.

Luigi Grippo and Marco Sciandrone. Nonmonotone globalization techniques for the barzilai-borwein gradient method. *Computational Optimization and Applications*, 23:143–169, 2002.

Toshimori Honjo, Tomohiro Sonobe, Kensuke Inaba, Takahiro Inagaki, , and Hiroki Takesue. 100,000-spin coherent ising machine. *Science Advances*, 7(40):eabh0952, 2021.

Roger A. Horn and Charles R. Johnson. *Matrix analysis*. Cambridge University Press, 1985.

Yuma Ichikawa and Yamato Arai. Optimization by parallel quasi-quantum annealing with gradient-based sampling. In *The Thirteenth International Conference on Learning Representations*, 2025.

T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, and H. Takesue. A coherent ising machine for 2000-node optimization problems. *Science*, 354:603–606, 2016.

S. V. Isakov, I. N. Zintchenko, T. F. Rønnow, and M. Troyer. Optimized simulated annealing for ising spin glasses. *Computer Physics Communications*, 192:265–271, 2015.

Ernst Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31:253–258, 1925.

Z. Jiang, G. Chen, R. Qiao, P. Feng, Y. Chen, J. Su, Z. Zhao, M. Jin, X. Chen, and H. Lu. Point convolutional neural network algorithm for Ising model ground state research based on spring vibration. *Scientific Reports*, 14:2643, 2024.

Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. *Physical Review E*, 58:5355–5363, Nov 1998.

Taro Kanao and Hayato Goto. Simulated bifurcation assisted by thermal fluctuation. *Communications Physics*, 5:153, 2022.

D. B. Kell. Scientific discovery as a combinatorial optimisation problem: how best to navigate the landscape of possible experiments? *Bioessays*, 34:236–244, 2012.

Andrew D. King, William Bernoudy, James King, Andrew J. Berkley, and Trevor Lanting. Emulating the coherent ising machine with a mean-field algorithm. *arXiv:1806.08422*, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Proceedings International Conference on Learning Representations (ICLR)*, 2015.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220: 671–680, 1983.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Hao Lo, William Moy, Hanzhao Yu, Sachin Sapatnekar, and Chris H. Kim. An ising solver chip based on coupled ring oscillators with a 48-node all-to-all connected array architecture. *Nature Electronics*, 6:771–778, 2023.

Stanisław Łojasiewicz. Une propriété topologique des sous-ensembles analytiques réels. In *Les Équations aux Dérivées Partielles, Colloques Internationaux du CNRS 117*, pp. 87–89, Paris, 1963. Éditions du Centre National de la Recherche Scientifique.

A. Lucas. Ising formulations of many NP problems. *Frontiers in Physics*, 2:5, 2014.

Zhi-Quan Tom Luo, Wing-Kin Ma, Anthony Man-Cho So, Yinyu Ye, and Shuzhong Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27:20–34, 2010.

Fuda Ma and Jin-Kao Hao. A multiple search operator heuristic for the max-k-cut problem. *Annals of Operations Research*, 248(1):365–403, 2017.

Olivier Maher, Manuel Jiménez, Corentin Delacour, Nele Harnack, Juan Núñez, María J. Avedillo, Bernabé Linares-Barranco, Aida Todri-Sanial, Giacomo Indiveri, and Siegfried Karg. A cmos-compatible oscillation-based VO_2 ising machine solver. *Nature Communications*, 15(3334), 2024.

Gisiro Maruyama. Continuous markov processes and stochastic equations. *Rendiconti del Circolo Matematico di Palermo*, 4(1):48–90, 1955.

M.L. Mehta. *Random matrices*, volume 142 of *Pure and applied mathematics*. Elsevier Academic Press, 3rd edition, 2004. ISBN 9780120884094.

Naeimeh Mohseni, Peter L. McMahon, and Tim Byrnes. Ising machines as hardware solvers of combinatorial optimization problems. *Nature Reviews Physics*, 4:363–379, 2022.

William Moy, Ibrahim Ahmed, Po-wei Chiu, John Moy, Sachin S. Sapatnekar, and Chris H. Kim. A 1,968-node coupled ring oscillator circuit for combinatorial optimization problem solving. *Nature Electronics*, 5:310–317, 2022.

Yurii Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods and Software*, 9(1–3):141–160, 1998.

Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer International Publishing, 2018.

Shaila Niazi, Shuvro Chowdhury, Navid Anjum Aadit, Masoud Mohseni, Yao Qin, and Kerem Y. Camsari. Training deep boltzmann networks with sparse ising machines. *Nature Electronics*, 7: 610–619, 2024.

Román Orús, Samuel Mugel, and Enrique Lizaso. Quantum computing for finance: overview and prospects. *Reviews in Physics*, 4:100028, 2019. ISSN 2405-4283.

D. N. Phan, H. M. Le, and H. A. Le Thi. Accelerated difference of convex functions algorithm and its application to sparse binary logistic regression. In *Proceedings of the Twenty Seventh International Joint Conference on Artificial Intelligence*, volume 18 of *IJCAI*, 2018.

Sayantan Pramanik, Sourav Chatterjee, and Harshkumar Oza. Convergence analysis of opto-electronic oscillator based coherent ising machines. *International Conference on COMmunication Systems & NETworkS*, pp. 1076–1081, 2024.

Shruti Puri, Christian Kraglund Andersen, Arne L. Grimsmo, and Alexandre Blais. Quantum annealing with all-to-all connected nonlinear oscillators. *Nature Communications*, 8(1):15785, 2017.

E. G. Rieffel, D. Venturelli, B. O'Gorman, M. B. Do, E. M. Prystay, and V. N. Smelyanskiy. A case study in programming a quantum annealer for hard operational planning problems. *Quantum Information Processing*, 14:1–36, 2015.

Anton Robert, Panagiotis Kl. Barkoutsos, Stefan Woerner, and Ivano Tavernelli. Resource-efficient quantum algorithm for protein folding. *npj Quantum Information*, 7:38, 2021.

Walter Rudin. *Principles of mathematical analysis*. McGraw-Hill, New York, 3rd edition, 1976. ISBN 9780070542358.

Yousef Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, second edition, 2003.

Giuseppe E. Santoro, Roman Martoňák, Erio Tosatti, and Roberto Car. Theory of quantum annealing of an ising spin glass. *Science*, 295(5564):2427–2430, 2002.

Zi-Song Shen, Feng Pan, Yao Wang, Yi-Ding Men, Wen-Biao Xu, Man-Hong Yung, and Pan Zhang. Free-energy machine for combinatorial optimization. *Nature Computational Science*, 5:322–332, 2025.

David Sherrington and Scott Kirkpatrick. Solvable model of a spin-glass. *Physical Review Letters*, 35:1792–1796, 1975.

Haoran Sun, Katayoon Goshvadi, Azade Nova, Dale Schuurmans, and Hanjun Dai. Revisiting sampling for combinatorial optimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32859–32874. PMLR, 2023.

Kotaro Tanahashi, Shinichi Takayanagi, Tomomitsu Motohashi, and Shu Tanaka. Application of ising machines and a software development for ising machines. *Journal of the Physical Society of Japan*, 88(6):061010, 2019.

Egor S. Tiunov, Alexander E. Ulanov, and A. I. Lvovsky. Annealing by simulating the coherent ising machine. *Optics Express*, 27(7):10288–10295, Apr 2019.

Máté Tibor Veszeli and Gábor Vattay. Mean field approximation for solving QUBO problems. *PLOS ONE*, 17(8):e0273709, 2022. doi: 10.1371/journal.pone.0273709.

Baonan Wang, Feng Hu, Haonan Yao, and Chao Wang. Prime factorization algorithm based on parameter optimization of ising model. *Scientific Reports*, 10:7106, 2020.

Juntao Wang, Daniel Ebler, K. Y. Michael Wong, David Shui Wing Hui, and Jie Sun. Bifurcation behaviors shape how continuous physical dynamics solves discrete ising optimization. *Nature Communications*, 14(1):2510, 2023.

Zhe Wang, Alireza Marandi, Kai Wen, Robert L Byer, and Yoshihisa Yamamoto. Coherent ising machine based on degenerate optical parametric oscillators. *Physical Review A*, 88(6):063853, 2013.

A. Wiegele. Biq mac library – a collection of max-cut and quadratic 0–1 programming instances of medium size. `https://biqmac.aau.at/biqmaclib.pdf`, 2007.

Yoshihisa Yamamoto, Kazuyuki Aihara, Timothee Leleu, Ken ichi Kawarabayashi, Satoshi Kako, Martin Fejer, Kyo Inoue, and Hiroki Takesue. Coherent ising machines-optical neural networks operating at the quantum limit. *npj Quantum Information*, 3(1):49, 2017.

A. N. Yavorsky, Liubov A. Markovich, Evgeny A Polyakov, and Alexey N. Rubtsov. Highly parallel algorithm for the ising ground state searching problem. *arXiv: Quantum Physics*, 2019.

Yuan Yuan, Ahmad Alabdulkareem, and Alex 'Sandy' Pentland. An interpretable approach for social network formation among heterogeneous agents. *Nature Communications*, 9(1):4704, 2018.

Nan Zhang, Shifei Ding, Jian Zhang, and Yu Xue. An overview on restricted boltzmann machines. *Neurocomputing*, 275:1186–1199, 2018. ISSN 0925-2312.

Appendix

This appendix provides a complete technical account of the proposed DOCH/ADOCH solvers.

We begin with an illustration of the Difference-of-Convex Hamiltonian on a two-spin Ising model in Appendix A and then reduce general Ising models to the homogeneous setting in Appendix B.

We next prove the mathematical results stated in Sections 2 and 3 (Appendix C), establish the relation of our attractor with physics-based nonlinear oscillator Hamiltonians (Appendix D), derive the DOCH algorithm 1 (Appendix E), and present the convergence analysis of DOCH and ADOCH (Appendix F).

We then discuss stability, boundedness, parameter selection and termination criteria with ablation studies (Appendix G), followed by the MAX-CUT/GW-SDP formulation (Appendix H) and implementation details of existing Ising solvers (Appendix I).

Finally, we detail ultra-large Ising data generation (Appendix J), scalable multi-GPU matrix-vector multiplication (Appendix K), hardware benchmarking setup (Appendix L), the matched benchmarking protocol and additional comparisons (Appendix M), and additional benchmarking figures (Appendix N).

A Continuous Relaxation and Landscape Synthesis of DOCH

The figure illustrates the continuous relaxation for a two-spin Ising problem. Discrete energy values are extended to a smooth landscape $\mathcal{E}$ and combined with the attractor $\mathcal{A}$ to form the differentiable Hamiltonian $\mathcal{H}$. After optimization, a continuous critical point is rounded to a binary spin vector. In this illustrative two-spin example the rounded vector is a ground state; this behavior is not guaranteed for a general interaction matrix.

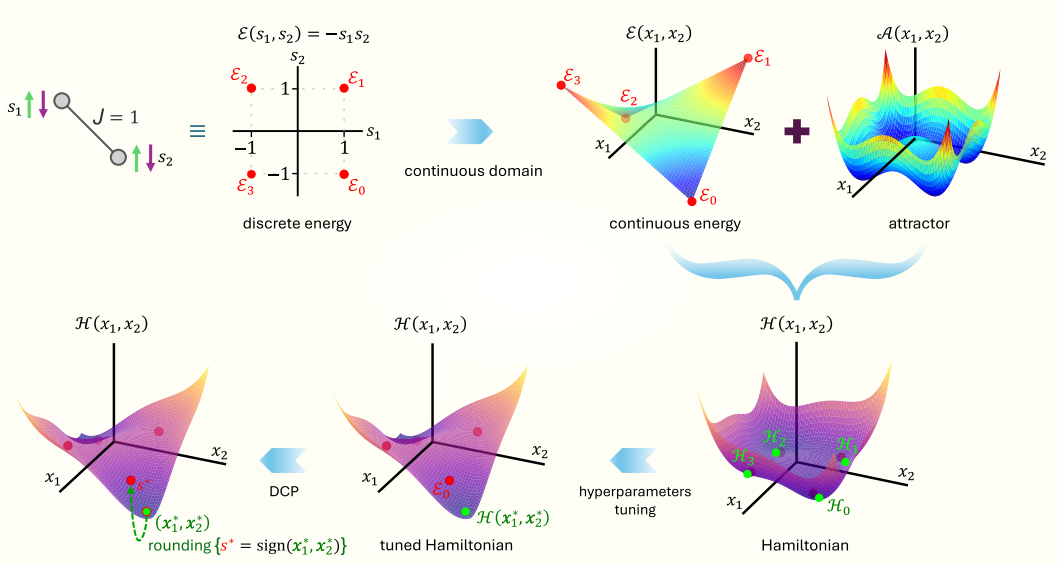


Figure 1: **Overview for a 2-spin system**. We relax the $2^2 = 4$ discrete energy values $(\mathcal{E}_0, \mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3)$ to a smooth energy landscape $\mathcal{E}(x_1, x_2)$ over $(x_1, x_2) \in \mathbb{R}^2$ and combine it with the attractor $\mathcal{A}(x_1, x_2)$ to form $\mathcal{H} = \mathcal{E} + \mathcal{A}$. Local minima of $\mathcal{H}$ are candidate rounded solutions. For this illustrative instance, $\text{sign}(x_1^*, x_2^*)$ agrees with a ground state $\boldsymbol{s}^*$; no general rounding guarantee is implied.

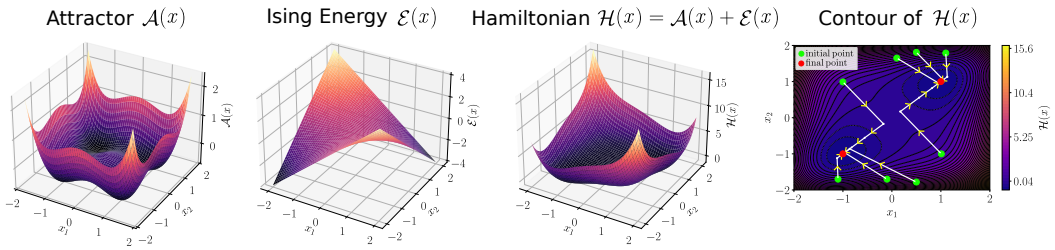


Figure 2: **DOCH trajectory for a 2-spin system.** We consider the antiferromagnetic model with $J_{12} = J_{21} = -1$, whose ground states are $(1, -1)$ and $(-1, 1)$. **First three columns**: Surface plots of the attractor $\mathcal{A}$, Ising energy $\mathcal{E}$, and Hamiltonian $\mathcal{H} = \mathcal{A} + \mathcal{E}$. **Last column**: Contour plot of $\mathcal{H}$ with $\alpha = 1$ and $\beta = 2$. DOCH trajectories (white curves) are shown for several initial points (green dots). In this example, all iterates converge to one of two degenerate ground states (red dots) within five steps; no general rounding guarantee is implied.

B Homogeneous Reformulation of Ising Models

We can represent any n-spin Ising model with an external field as an $(n+1)$-spin Ising model without an external field. Consider the Ising model with an external field:

$$\mathcal{E}(\boldsymbol{s}) = -\frac{1}{2}\boldsymbol{s}^\top \mathbf{J}\boldsymbol{s} - \boldsymbol{h}^\top \boldsymbol{s} \qquad \big(\boldsymbol{s} \in \{-1, 1\}^n\big), \tag{17}$$

where $\mathbf{J}$ is the coupling matrix and $\boldsymbol{h}$ is the external field vector. Now, define the coupling matrix and energy function

$$\hat{\mathbf{J}} = \begin{pmatrix} \mathbf{J} & \boldsymbol{h} \\ \boldsymbol{h}^\top & 0 \end{pmatrix}, \qquad \hat{\mathcal{E}}(\boldsymbol{\sigma}) = -\frac{1}{2}\boldsymbol{\sigma}^\top \hat{\mathbf{J}}\boldsymbol{\sigma} \qquad \big(\boldsymbol{\sigma} \in \{-1, 1\}^{n+1}\big). \tag{18}$$

Proposition B.1. *Suppose $\boldsymbol{\sigma}^* = (\boldsymbol{s}_0, t_0)$ is the ground state of* (18)*, where $\boldsymbol{s}_0 \in \{-1, +1\}^n$ and $t_0 \in \{-1, 1\}$. Then $\boldsymbol{s}^* = t_0\boldsymbol{s}_0 \in \{-1, +1\}^n$ is the ground state of* (17)*.*

Proof. Let $\boldsymbol{s} \in \{-1, 1\}^n$ and $t \in \{-1, 1\}$. From (17) and (18),

$$\hat{\mathcal{E}}\left((\boldsymbol{s}, t)\right) = \mathcal{E}(t\boldsymbol{s}). \tag{19}$$

On the other hand, by the definition of $\boldsymbol{\sigma}^*$,

$$\hat{\mathcal{E}}(\boldsymbol{\sigma}^*) = \hat{\mathcal{E}}\left((\boldsymbol{s}_0, t_0)\right) \leqslant \hat{\mathcal{E}}\left((\boldsymbol{s}, t)\right). \tag{20}$$

Comparing (19) and (20), we obtain

$$\mathcal{E}(\boldsymbol{s}^*) = \mathcal{E}(t_0\boldsymbol{s}_0) = \hat{\mathcal{E}}((\boldsymbol{s}_0, t_0)) \leqslant \hat{\mathcal{E}}((\boldsymbol{s}, 1)) = \mathcal{E}(\boldsymbol{s})$$

Since this holds for any $\boldsymbol{s} \in \{-1, 1\}^n$, it follows that $\boldsymbol{s}^*$ is a ground state of (17). □

C Mathematical Results

In this section, we establish the proof of the mathematical results (Propositions 2.1–3.1) described in Section 2 and 3 of the manuscript. These results will be useful for the convergence analysis of our iterative solver in Appendix F.

C.1 Proof of Proposition 2.1

Statement: The global minimizers of $\mathcal{A}$ are exactly the points $\{-\lambda, +\lambda\}^n$, where $\lambda = \sqrt{\alpha/\beta}$.

Proof. The gradient of the attractor $\nabla\mathcal{A}(\boldsymbol{x}) \in \mathbb{R}^n$ has components

$$\nabla\mathcal{A}(\boldsymbol{x})_i = \beta x_i^3 - \alpha x_i \qquad (i = 1, \ldots, n), \tag{21}$$

and its Hessian $\nabla^2\mathcal{A}(\boldsymbol{x}) \in \mathbb{R}^{n\times n}$ is a diagonal matrix with

$$\nabla^2\mathcal{A}(\boldsymbol{x})_{ii} = 3\beta x_i^2 - \alpha \qquad (i = 1, \ldots, n). \tag{22}$$

We can conclude from (21) that the critical points of $\mathcal{A}$ are exactly the points $\{-\lambda, 0, \lambda\}^n$, where $\lambda = \sqrt{\alpha/\beta}$. This is because $\nabla\mathcal{A}(\boldsymbol{x}) = \mathbf{0}$ if and only if each $x_i \in \{-\lambda, 0, \lambda\}$. Now, a sufficient condition for a critical point $\boldsymbol{x}$ to be a local minimizer is that $\nabla^2\mathcal{A}(\boldsymbol{x})_{ii} > 0$ for all $i = 1, \ldots, n$. It follows from (22) that this condition holds if each $x_i \in \{-\lambda, \lambda\}$. On the other hand, if $x_i = 0$ for some $i = 1, \ldots, n$, then $\nabla^2\mathcal{A}(\boldsymbol{x})_{ii} < 0$, so that $\boldsymbol{x}$ cannot be a local minimizer. Thus, the local minimizer of $\mathcal{A}$ are exactly the points $\{-\lambda, +\lambda\}^n$. Finally, observe that $\mathcal{A}$ attains the same value at each of these points. Therefore, these points are exactly the global minimizers of $\mathcal{A}$. □

C.2 Proof of Proposition 2.2

Statement: The Hamiltonian $\mathcal{H}$ is coercive, bounded below on $\mathbb{R}^n$ and has a minimizer.

Proof. We have

$$\sum_{i=1}^{n} x_i^4 \geqslant \frac{1}{n^2}\Big(\sum_{i=1}^{n} x_i^2\Big)^2 = \frac{1}{n^2}\|\boldsymbol{x}\|_2^4,$$

where $\|\boldsymbol{x}\|_2$ is the Euclidean norm of $\boldsymbol{x}$. This gives us the lower bound

$$\frac{\beta}{4}\sum_{i=1}^{n} x_i^4 \geqslant \frac{\beta}{4n^2}\|\boldsymbol{x}\|_2^4.$$

On the other hand, we have the upper bound

$$\frac{1}{2}\boldsymbol{x}^\top(\mathbf{J} + \alpha\mathbf{I})\boldsymbol{x} \leqslant \frac{1}{2}c\|\boldsymbol{x}\|_2^2,$$

where $c = \lambda_{\max}(\mathbf{J} + \alpha\mathbf{I}) > 0$. Combining the above bounds, we have

$$\mathcal{H}(\boldsymbol{x}) \geqslant \frac{\beta}{4n^2}\|\boldsymbol{x}\|_2^4 - \frac{1}{2}c\|\boldsymbol{x}\|_2^2 = \frac{1}{4n^2}\|\boldsymbol{x}\|_2^2\Big(\beta\|\boldsymbol{x}\|_2^2 - 2n^2c\Big).$$

As $\beta > 0$, the right side goes to $+\infty$ as $\|\boldsymbol{x}\|_2 \to \infty$. Hence, $\mathcal{H}$ is coercive. Also,

$$\frac{1}{4n^2}\|\boldsymbol{x}\|_2^2\Big(\beta\|\boldsymbol{x}\|_2^2 - 2n^2c\Big) \geqslant -\frac{n^2c^2}{4\beta},$$

which implies $\mathcal{H}$ is bounded below. So, applying Weierstrass' extreme value theorem Beck (2017) and continuity of $\mathcal{H}$, we can conclude the existence of a minimizer $\boldsymbol{x}^*$. □

C.3 Proof of Proposition 2.3

Statement: Suppose $\boldsymbol{x}^*$ is a minimizer of $\mathcal{H}$ and it lies on a vertex of the scaled hypercube $\{-\lambda, \lambda\}^n$, where $\lambda > 0$. Then $\operatorname{sign}(\boldsymbol{x}^*)$ is the ground state of the Ising model (2).

Proof. Let $\boldsymbol{s}^* \in \{-1, 1\}^n$ be a minimizer of (2). By optimality, $\mathcal{E}(\boldsymbol{s}^*) \leqslant \mathcal{E}(\operatorname{sign}(\boldsymbol{x}^*))$. Hence, we just have to show that

$$\mathcal{E}(\operatorname{sign}(\boldsymbol{x}^*)) \leqslant \mathcal{E}(\boldsymbol{s}^*). \tag{23}$$

By assumption, for all $\boldsymbol{x} \in \mathbb{R}^n$,

$$\mathcal{A}(\boldsymbol{x}^*) + \mathcal{E}(\boldsymbol{x}^*) = \mathcal{H}(\boldsymbol{x}^*) \leqslant \mathcal{H}(\boldsymbol{x}) = \mathcal{A}(\boldsymbol{x}) + \mathcal{E}(\boldsymbol{x}). \tag{24}$$

Since $\boldsymbol{x}^* \in \{-\lambda, +\lambda\}^n$, we have $\boldsymbol{x}^* = \lambda\operatorname{sign}(\boldsymbol{x}^*)$. Setting $\boldsymbol{x}^* = \lambda\operatorname{sign}(\boldsymbol{x}^*)$ and $\boldsymbol{x} = \lambda\boldsymbol{s}^*$ in (24) gives

$$\mathcal{A}(\lambda\operatorname{sign}(\boldsymbol{x}^*)) + \mathcal{E}(\lambda\operatorname{sign}(\boldsymbol{x}^*)) \leqslant \mathcal{A}(\lambda\boldsymbol{s}^*) + \mathcal{E}(\lambda\boldsymbol{s}^*). \tag{25}$$

From (2) and (4), it follows that $\mathcal{E}(\lambda\boldsymbol{x}) = \lambda^2\mathcal{E}(\boldsymbol{x})$ and $\mathcal{A}(\lambda\operatorname{sign}(\boldsymbol{x}^*)) = \mathcal{A}(\lambda\boldsymbol{s}^*)$. Substituting these into (25), we get (23). □

C.4 Proof of Proposition 3.1

Statement: The function f is convex for all $\beta > 0$, and there exists α_0 such that g is strongly convex for all $\alpha > \alpha_0$.

Proof. Both f and g are twice differentiable, and their Hessians are

$$\nabla^2 f(\boldsymbol{x}) = 3\beta \begin{pmatrix} x_1^2 & & \\ & \ddots & \\ & & x_n^2 \end{pmatrix} \quad \text{and} \quad \nabla^2 g(\boldsymbol{x}) = \mathbf{J} + \alpha \mathbf{I}.$$

Clearly, $\nabla^2 f(\boldsymbol{x})$ is positive semidefinite for all $\beta > 0$, which establishes the convexity of f.

On the other hand, $\nabla^2 g(\boldsymbol{x})$ is positive definite if and only if $\lambda_{\min}(\mathbf{J} + \alpha \mathbf{I}) > 0$. As $\mathbf{J}$ is symmetric, it has real eigenvalues. Moreover, $J_{ii} = 0$ implies $\operatorname{tr}(\mathbf{J}) = 0$, and hence $\lambda_{\min}(\mathbf{J}) \leqslant 0$; the inequality is strict whenever $\mathbf{J} \neq \mathbf{0}$. Defining

$$\alpha_0 = \lambda_{\max}(-\mathbf{J}) = -\lambda_{\min}(\mathbf{J}) \geqslant 0,$$

we have $\lambda_{\min}(\mathbf{J} + \alpha \mathbf{I}) > 0$ for every $\alpha > \alpha_0$. Thus, g is convex at $\alpha = \alpha_0$ and strongly convex for $\alpha > \alpha_0$. □

D Relation with KPO and OPO

We show that for certain parameter settings and under some assumptions, the Hamiltonians in Goto (2018; 2019) resemble our Hamiltonian.

We form the Hamiltonian by combining the Ising energy $\mathcal{E}$ with the attractor $\mathcal{A}$:

$$\mathcal{H} = \mathcal{A} + \mathcal{E}. \tag{26}$$

The key observation behind our algorithm is that (26) can be written as a difference of convex functions, $\mathcal{H} = f - g$.

The Hamiltonian for the Kerr-nonlinear parametric oscillator (KPO) is of the form

$$\begin{aligned} \mathcal{H}_c(\boldsymbol{x}, \boldsymbol{y}, t) = \sum_{i=1}^{N} \left(\frac{K}{4}(x_i^2 + y_i^2)^2 - \frac{p(t)}{2}(x_i^2 - y_i^2) + \frac{\Delta_i}{2}(x_i^2 + y_i^2) \right) \\ - \frac{\xi_0}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} J_{ij}(x_i x_j + y_i y_j), \end{aligned} \tag{27}$$

where ξ_0 is a positive constant, K is the Kerr coefficient, $p(t)$ is the time-dependent pumping amplitude, Δ_i is the detuning frequency of the i-th oscillator, x_i is the state of the i-th spin, and y_i is its momentum. On setting $y_i = 0, \Delta_i = \Delta$ in (27), we get

$$\begin{aligned} \mathcal{H}_{KPO}(\boldsymbol{x}, \boldsymbol{y}, t) &= \sum_{i=1}^{N} \left(\frac{K}{4} x_i^4 - \frac{p(t) - \Delta_i}{2} x_i^2 \right) - \frac{\xi_0}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} J_{ij} x_i x_j \\ &= \frac{K}{4}(x_1^4 + \cdots + x_n^4) - \frac{1}{2}\big(p(t) - \Delta\big)(x_1^2 + \cdots + x_n^2) - \frac{\xi_0}{2} \boldsymbol{x}^\top \mathbf{J} \boldsymbol{x}. \end{aligned} \tag{28}$$

Thus, we can identify $\xi_0^{-1}\mathcal{H}_{KPO}$ with our Hamiltonian $\mathcal{H} = f - g$, where $\beta = K\xi_0^{-1}$ and $\alpha = \xi_0^{-1}(p(t) - \Delta)$ in (32). Similarly, the Hamiltonian for the optical parametric oscillators (OPO) can be written as

$$\mathcal{H}_{OPO} = \sum_{i=1}^{N} \left(\frac{\kappa_2}{4}(x_i^2 + y_i^2)^2 - \frac{p}{2}(x_i^2 - y_i^2) + \frac{\kappa_1}{2}(x_i^2 + y_i^2) \right) - \frac{\xi_0}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} J_{i,j}(x_i x_j + y_i y_j) \tag{29}$$

with κ_1 and κ_2 being the one-photon and two-photon loss rates. As in the reduction for the KPO, a scaled version of $\mathcal{H}_{\text{OPO}}$ can be identified with our Hamiltonian by setting $\alpha = \xi_0^{-1}(p - \kappa_1)$ and $\beta = \kappa_2 \xi_0^{-1}$ in (32). Although the structure of our Hamiltonian resembles (28) and (29), it is important to note that the latter cannot generally be expressed as a difference of convex functions.

E Derivation of DOCH Algorithm

We obtain the Difference-of-Convex Hamiltonian (DOCH) Ising solver by applying the Difference-of-Convex Algorithm (DCA) to our problem (6), resulting in a simple iterative scheme. Starting from an initial point $\boldsymbol{x}^{(k)}$, we linearize g using its first-order approximation around $\boldsymbol{x}^{(k)}$. This yields a convex surrogate, which is minimized to obtain the next iterate $\boldsymbol{x}^{(k+1)}$. In particular, as g is convex, its linear approximation at $\boldsymbol{x}^{(k)}$ gives us a global lower bound Beck (2017). Specifically, for all $\boldsymbol{x} \in \mathbb{R}^n$, we have

$$g(\boldsymbol{x}) \geqslant g(\boldsymbol{x}^{(k)}) + \nabla g(\boldsymbol{x}^{(k)})^\top (\boldsymbol{x} - \boldsymbol{x}^{(k)}).$$

This results in the following convex upper bound on the Hamiltonian:

$$\mathcal{H}(\boldsymbol{x}) = f(\boldsymbol{x}) - g(\boldsymbol{x}) \leqslant f(\boldsymbol{x}) - g(\boldsymbol{x}^{(k)}) - \nabla g(\boldsymbol{x}^{(k)})^\top (\boldsymbol{x} - \boldsymbol{x}^{(k)}).$$

We refine our current estimate $\boldsymbol{x}^{(k)}$ by minimizing this upper bound. Specifically, we set the update $\boldsymbol{x}^{(k+1)}$ as the minimizer of the surrogate function:

$$F(\boldsymbol{x}) = f(\boldsymbol{x}) - g(\boldsymbol{x}^{(k)}) - \nabla g(\boldsymbol{x}^{(k)})^\top (\boldsymbol{x} - \boldsymbol{x}^{(k)}).$$

As F is convex and differentiable, by first-order optimality,

$$\mathbf{0} = \nabla F(\boldsymbol{x}^{(k+1)}) = \nabla f(\boldsymbol{x}^{(k+1)}) - \nabla g(\boldsymbol{x}^{(k)}). \tag{30}$$

Substituting $\nabla g(\boldsymbol{x}) = (\mathbf{J} + \alpha \mathbf{I})\boldsymbol{x}$ in (30) and after some calculation, we get

$$\boldsymbol{x}^{(k+1)} = \mathcal{T}(\boldsymbol{x}^{(k)}), \tag{31}$$

where

$$\mathcal{T}(\boldsymbol{x}) = \varphi\big(\beta^{-1}(\mathbf{J} + \alpha \mathbf{I})\boldsymbol{x}\big),$$

and

$$\varphi(x_1, \ldots, x_n) = (\sqrt[3]{x}_1, \ldots, \sqrt[3]{x}_n).$$

Thus, $\mathcal{T}$ is a linear transform followed by a componentwise cube root operation. The resulting algorithm, called Difference-of-Convex Hamiltonian (DOCH), is summarized in Algorithm 1.

F Convergence Analysis

We present a self-contained convergence analysis of DOCH. Recall that DOCH iteratively minimizes the Hamiltonian $\mathcal{H} = f - g$ using DCA, where

$$f(\boldsymbol{x}) = \frac{\beta}{4}\left(x_1^4 + \cdots + x_n^4\right) \quad \text{and} \quad g(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top (\mathbf{J} + \alpha \mathbf{I})\boldsymbol{x}. \tag{32}$$

Following Proposition 3.1, we assume that g is strongly convex, and

$$\mu = \lambda_{\min}(\mathbf{J} + \alpha \mathbf{I}) > 0. \tag{33}$$

F.1 Proof of Proposition 4.1

Statement: Let $\{\boldsymbol{x}^{(k)}\}$ be the iterates generated by DOCH. Then

$$\mathcal{H}\big(\boldsymbol{x}^{(k)}\big) - \mathcal{H}\big(\boldsymbol{x}^{(k+1)}\big) \geqslant \frac{\mu}{2}\,\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|^2. \tag{34}$$

Proof. By construction, $\boldsymbol{x}^{(k+1)}$ is the minimizer of the function

$$F(\boldsymbol{x}) = f(\boldsymbol{x}) - \big(g(\boldsymbol{x}^k) + \nabla g(\boldsymbol{x}^{(k)})^\top (\boldsymbol{x} - \boldsymbol{x}^{(k)})\big). \tag{35}$$

Therefore,

$$f(\boldsymbol{x}^{(k+1)}) - g(\boldsymbol{x}^k) - \nabla g(\boldsymbol{x}^{(k)})^\top (\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}) = F(\boldsymbol{x}^{(k+1)}) \leqslant F(\boldsymbol{x}^{(k)}) = f(\boldsymbol{x}^{(k)}) - g(\boldsymbol{x}^k),$$

giving us

$$f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^{(k+1)}) \geqslant \nabla g(\boldsymbol{x}^{(k)})^\top (\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k+1)}). \tag{36}$$

As $\mathcal{H} = f - g$, we have from (36) that

$$\begin{aligned}\mathcal{H}\big(\boldsymbol{x}^{(k)}\big) - \mathcal{H}\big(\boldsymbol{x}^{(k+1)}\big) &= f(\boldsymbol{x}^{(k)}) - g(\boldsymbol{x}^{(k)}) - \big(f(\boldsymbol{x}^{(k+1)}) - g(\boldsymbol{x}^{(k+1)})\big) \\ &\geqslant g(\boldsymbol{x}^{(k+1)}) - g(\boldsymbol{x}^{(k)}) - \nabla g(\boldsymbol{x}^{(k)})^\top(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}). \end{aligned} \tag{37}$$

Substituting the formula of g and ∇g (see (32)) in (37), we get

$$\mathcal{H}\big(\boldsymbol{x}^{(k)}\big) - \mathcal{H}\big(\boldsymbol{x}^{(k+1)}\big) \geqslant \frac{1}{2}(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)})^\top(\mathbf{J} + \alpha\mathbf{I})(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}).$$

Moreover, from (33),

$$(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)})^\top(\mathbf{J} + \alpha\mathbf{I})(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}) \geqslant \mu\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k+1)}\|^2.$$

This establishes the desired result (34). □

In general, the challenging part of convergence analysis is to show that the iterates remain bounded. This is often assumed in the analysis Phan et al. (2018). However, for DOCH, boundedness of the iterates can be established unconditionally.

F.2 Proof of Proposition 4.2

Statement: The DOCH iterates are bounded, and

$$\lim_{k\to\infty} \big\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\big\|_2 = 0. \tag{38}$$

Proof. By Proposition 4.1, the sequence $\{\mathcal{H}(\boldsymbol{x}^{(k)})\}$ is monotone (nonincreasing). Hence, for all $k \geqslant 1$,

$$\mathcal{H}(\boldsymbol{x}^{(k)}) \leqslant \mathcal{H}(\boldsymbol{x}^{(0)}). \tag{39}$$

As $\mathcal{H}$ is coercive, the sublevel set $\{\boldsymbol{x} : \mathcal{H}(\boldsymbol{x}) \leqslant \mathcal{H}(\boldsymbol{x}^{(0)})\}$ is bounded. However, by (39), the sequence $\{\boldsymbol{x}^{(k)}\}$ is contained in this sublevel set and is hence bounded.

On the other hand, summing the descent inequality (34) from $k = 0$ to N gives

$$\mathcal{H}(\boldsymbol{x}^{(0)}) - \mathcal{H}(\boldsymbol{x}^{(N+1)}) = \sum_{k=0}^{N}\Big(\mathcal{H}\big(\boldsymbol{x}^{(k)}\big) - \mathcal{H}\big(\boldsymbol{x}^{(k+1)}\big)\Big) \geqslant \frac{\mu}{2}\sum_{k=0}^{N}\big\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k+1)}\big\|_2^2.$$

Since $\mathcal{H}(\boldsymbol{x}^{(k)})$ is bounded below, there exists $C \in \mathbb{R}$ such that $\mathcal{H}(\boldsymbol{x}^{(k)}) \geqslant C$ for all $k \geqslant 1$. Hence,

$$\frac{\mu}{2}\sum_{k=0}^{N}\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k+1)}\|_2^2 \leqslant \mathcal{H}(\boldsymbol{x}^{(0)}) - \mathcal{H}(\boldsymbol{x}^{(N+1)}) \leqslant \mathcal{H}(\boldsymbol{x}^{(0)}) - C < +\infty.$$

Letting $N \to \infty$ gives

$$\frac{\mu}{2}\sum_{k=0}^{\infty}\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k+1)}\|_2^2 < +\infty. \tag{40}$$

Consequently, we obtain (38). □

Note that (40) by itself does not guarantee that the sequence $\{\boldsymbol{x}^{(k)}\}$ converges. To prove convergence, we must show that $\{\boldsymbol{x}^{(k)}\}$ is a Cauchy sequence Rudin (1976). This will be done in Theorem 4.5, using the fact that the Hamiltonian $\mathcal{H}$ is real-analytic (being a polynomial) and therefore satisfies the Łojasiewicz gradient inequality Łojasiewicz (1963). Based on the analysis so far, we can already state the following result.

F.3 Proof of Proposition 4.4

Statement: All limit points of the DOCH iterates are critical points of $\mathcal{H}$.

Proof. The iterates $\{\boldsymbol{x}^{(k)}\}$ are bounded by Proposition 4.2, and hence, by the Bolzano-Weierstrass theorem Rudin (1976), have at least one limit point, say, $\boldsymbol{x}^* \in \mathbb{R}^n$. In particular, we can extract a subsequence $\{\boldsymbol{x}^{(k_i)}\}$ such that

$$\lim_{i\to\infty} \boldsymbol{x}^{(k_i)} = \boldsymbol{x}^*. \tag{41}$$

We claim that $\boldsymbol{x}^*$ is a critical point of $\mathcal{H}$, i.e., $\nabla\mathcal{H}(\boldsymbol{x}^*) = \mathbf{0}$. Indeed, since $\boldsymbol{x}^{(k+1)}$ is a minimizer of (35), we have by first-order optimality that

$$\mathbf{0} = \nabla F(\boldsymbol{x}^{(k+1)}) = \nabla f(\boldsymbol{x}^{(k+1)}) - \nabla g(\boldsymbol{x}^{(k)}).$$

Thus, for all $k \geqslant 1$,

$$\nabla f(\boldsymbol{x}^{(k+1)}) = \nabla g(\boldsymbol{x}^{(k)}) \tag{42}$$

In particular,

$$\nabla\mathcal{H}(\boldsymbol{x}^{(k_i)}) = \nabla f(\boldsymbol{x}^{(k_i)}) - \nabla g(\boldsymbol{x}^{(k_i)}) = \nabla f(\boldsymbol{x}^{(k_i)}) - \nabla f(\boldsymbol{x}^{(k_i+1)}). \tag{43}$$

Now, by the triangle inequality,

$$\|\boldsymbol{x}^{(k_i+1)} - \boldsymbol{x}^*\|_2 \leqslant \|\boldsymbol{x}^{(k_i+1)} - \boldsymbol{x}^{(k_i)}\|_2 + \|\boldsymbol{x}^{(k_i)} - \boldsymbol{x}^*\|_2.$$

Letting $i \to \infty$ on both sides using (40) gives

$$\lim_{i\to\infty} \boldsymbol{x}^{(k_i+1)} = \boldsymbol{x}^*. \tag{44}$$

Finally, letting $i \to \infty$ in (43), we obtain $\nabla\mathcal{H}(\boldsymbol{x}^*) = \mathbf{0}$ from (41) and (44). □

We now use the classical Łojasiewicz gradient inequality Łojasiewicz (1963) to show that the DOCH iterates actually converge, that is, they admit a unique limit point. For this, we need the following result.

F.4 Proof of Proposition 4.3

Statement: There exists $\sigma > 0$ such that for all $k \geqslant 1$,

$$\|\nabla\mathcal{H}(\boldsymbol{x}^{(k)})\|_2 \leqslant \sigma\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\|_2. \tag{45}$$

Proof. From (42), we have

$$\nabla\mathcal{H}(\boldsymbol{x}^{(k)}) = \nabla f(\boldsymbol{x}^{(k)}) - \nabla g(\boldsymbol{x}^{(k)}) = \nabla g(\boldsymbol{x}^{(k-1)}) - \nabla g(\boldsymbol{x}^{(k)}).$$

In particular, since $\nabla g(\boldsymbol{x}) = (\mathbf{J} + \alpha\mathbf{I})\boldsymbol{x}$,

$$\|\nabla\mathcal{H}(\boldsymbol{x}^{(k)})\|_2 \leqslant \|(\mathbf{J} + \alpha\mathbf{I})(\boldsymbol{x}^{(k-1)} - \boldsymbol{x}^{(k)})\|_2.$$

Letting $\sigma = \lambda_{\max}(\mathbf{J} + \alpha\mathbf{I}) > 0$, we get (45). □

The following is the main convergence result. The proof follows the analysis in Absil et al. (2005); Attouch et al. (2013).

F.5 Proof of Theorem 4.5

Statement: The DOCH iterates $\{\boldsymbol{x}^{(k)}\}$ converge to a critical point of $\mathcal{H}$.

Proof. We know that $\{\boldsymbol{x}^{(k)}\}$ is bounded and hence has at least one limit point $\boldsymbol{x}^* \in \mathbb{R}^n$. Moreover, we know from Proposition 4.4 that $\boldsymbol{x}^*$ is a critical point of $\mathcal{H}$. To complete the proof, we just have to show that the entire sequence $\{\boldsymbol{x}^{(k)}\}$ converges to $\boldsymbol{x}^*$.

From Proposition 4.1, we know that $\{\mathcal{H}(\boldsymbol{x}^{(k)})\}$ is non-increasing. Moreover, since $\boldsymbol{x}^*$ is a limit point of $\boldsymbol{x}^{(k)}$ and $\mathcal{H}$ is continuous,

$$\lim_{k\to\infty} \mathcal{H}(\boldsymbol{x}^{(k)}) = \mathcal{H}(\boldsymbol{x}^*). \tag{46}$$

Let

$$\Delta_k = \mathcal{H}(\boldsymbol{x}^{(k)}) - \mathcal{H}(\boldsymbol{x}^*) \qquad \text{and} \qquad \delta_k = \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\|.$$

We can assume that $\Delta_k > 0$ and $\delta_k > 0$ for all $k \geqslant 1$. Indeed, if $\Delta_K = 0$ for some $K \geqslant 1$, then $\mathcal{H}(\boldsymbol{x}^{(k)}) = \mathcal{H}(\boldsymbol{x}^{(k+1)})$ for all $k \geqslant K$. Consequently, we have from (34) that $\delta_k = 0$ for all $k > K$. In other words, if Δ_K or δ_k vanishes for some K, then the sequence $\{\boldsymbol{x}^{(k)}\}$ becomes eventually constant, and therefore must converge to the limit point $\boldsymbol{x}^*$.

Since $\mathcal{H}$ is a polynomial, it satisfies the Łojasiewicz gradient inequality Łojasiewicz (1963); Absil et al. (2005) in some neighbourhood of the critical point $\boldsymbol{x}^*$. More specifically, there exist $\theta \in [0, 1), c > 0$ and $r > 0$, such that for all $\|\boldsymbol{x} - \boldsymbol{x}^*\| < r$,

$$|\mathcal{H}(\boldsymbol{x}) - \mathcal{H}(\boldsymbol{x}^*)|^\theta \leqslant c\,\|\nabla\mathcal{H}(\boldsymbol{x})\|_2, \tag{47}$$

We can conveniently reformulate this in terms of the function

$$\varphi(t) = \frac{c}{1-\theta}\, t^{1-\theta}. \tag{48}$$

We can verify that (48) is concave and non-decreasing on $[0, \infty)$, and $\varphi'(t) = ct^{-\theta}$ for any $t > 0$. Therefore, we can write (47) as

$$\varphi'\big(|\mathcal{H}(\boldsymbol{x}) - \mathcal{H}(\boldsymbol{x}^*)|\big)\, \big\|\nabla\mathcal{H}(\boldsymbol{x})\big\|_2 \geqslant 1. \tag{49}$$

We know from (38) and (46) that $\delta_k \to 0$ and $\Delta_k \to 0$. Since φ is non-decreasing and satisfies $\varphi(t) \to 0$ as $t \to 0$, it follows that $\varphi(\Delta_k) \to 0$. Moreover, since $\boldsymbol{x}^*$ is a limit point of $\{\boldsymbol{x}^{(k)}\}$, there exists $k_0 \geqslant 1$ such that

$$\|\boldsymbol{x}^{(k_0)} - \boldsymbol{x}^*\| < \frac{r}{2} \qquad \text{and} \qquad \frac{2\sigma}{\mu}\varphi(\Delta_{k_0}) + \delta_{k_0} < \frac{r}{2}. \tag{50}$$

In particular, we have from (49) that $\varphi'(\Delta_{k_0})\|\nabla\mathcal{H}(\boldsymbol{x}^{(k_0)})\|_2 \geqslant 1$. Combining this with Proposition 4.3, we get

$$\varphi'\big(\Delta_{k_0}\big) \geqslant \frac{1}{\sigma\delta_{k_0}}. \tag{51}$$

On the other hand, by (34), we have

$$\Delta_{k_0} - \Delta_{k_0+1} = \mathcal{H}\big(\boldsymbol{x}^{(k_0)}\big) - \mathcal{H}\big(\boldsymbol{x}^{(k_0+1)}\big) \geqslant \frac{\mu}{2}\,\delta_{k_0+1}^2. \tag{52}$$

Since φ is concave and differentiable on $(0, \infty)$,

$$\varphi(\Delta_{k_0}) - \varphi(\Delta_{k_0+1}) \geqslant \varphi'(\Delta_{k_0})(\Delta_{k_0} - \Delta_{k_0+1}).$$

From (51) and (52), we obtain

$$\varphi(\Delta_{k_0}) - \varphi(\Delta_{k_0+1}) \geqslant \frac{\mu}{2}\,\varphi'(\Delta_{k_0})\,\delta_{k_0+1}^2 \geqslant \frac{\mu}{2\sigma}\,\left(\frac{\delta_{k_0+1}^2}{\delta_{k_0}}\right).$$

In other words,

$$\delta_{k_0+1}^2 \leqslant \frac{2\sigma}{\mu}\big(\varphi(\Delta_{k_0}) - \varphi(\Delta_{k_0+1})\big)\,\delta_{k_0}.$$

Applying the AM-GM inequality, we get

$$2\delta_{k_0+1} \leqslant \frac{2\sigma}{\mu}\big(\varphi(\Delta_{k_0}) - \varphi(\Delta_{k_0+1})\big) + \delta_{k_0}. \tag{53}$$

Combining this with (50) gives

$$\delta_{k_0+1} \leqslant \frac{2\sigma}{\mu}\big(\varphi(\Delta_{k_0}) - \varphi(\Delta_{k_0+1})\big) + \delta_{k_0} \leqslant \frac{2\sigma}{\mu}\varphi(\Delta_{k_0}) + \delta_{k_0} \leqslant \frac{r}{2}. \tag{54}$$

From (50) and (54), we have

$$\|\boldsymbol{x}^{k_0+1} - \boldsymbol{x}^*\|_2 \leqslant \|\boldsymbol{x}^{k_0+1} - \boldsymbol{x}^{k_0}\|_2 + \|\boldsymbol{x}^{k_0} - \boldsymbol{x}^*\|_2 = \delta_{k_0+1} + \|\boldsymbol{x}^{k_0} - \boldsymbol{x}^*\|_2 < r.$$

Thus $\boldsymbol{x}^{k_0+1}$ is in the neighbourhood of $\boldsymbol{x}^*$ stipulated in (47). In fact, we claim that the entire tail $\{\boldsymbol{x}^k\}, k > k_0$, belongs to this neighbourhood. This can be shown by contradiction. Indeed, suppose k_1 is the smallest $k > k_0$ such that $\|\boldsymbol{x}^{(k_1)} - \boldsymbol{x}^*\|_2 \geqslant r$. This means that the points $\boldsymbol{x}^{(k_0)}, \ldots, \boldsymbol{x}^{(k_1-1)}$ are in the neighbourhood stipulated in (47). Applying (53) repeatedly, we have

$$\delta_{k+1} \leqslant \frac{2\sigma}{\mu}\big(\varphi(\Delta_k) - \varphi(\Delta_{k+1})\big) + (\delta_k - \delta_{k+1}), \qquad (k_0 \leqslant k < k_1).$$

Summing both sides, we obtain

$$\sum_{k=k_0}^{k_1-1} \delta_{k+1} \leqslant \frac{2\sigma}{\mu}\big(\varphi(\Delta_{k_0}) - \varphi(\Delta_{k_1})\big) + (\delta_{k_0} - \delta_{k_1}). \tag{55}$$

In particular,

$$\sum_{k=k_0}^{k_1-1} \delta_{k+1} \leqslant \frac{2\sigma}{\mu}\varphi(\Delta_{k_0}) + \delta_{k_0} < \frac{r}{2},$$

and

$$\|\boldsymbol{x}^{(k_1)} - \boldsymbol{x}^*\|_2 \leqslant \|\boldsymbol{x}^{(k_0)} - \boldsymbol{x}^*\|_2 + \sum_{k=k_0}^{k_1-1} \delta_{k+1} < r,$$

which contradicts our assumption about $\boldsymbol{x}^{(k_1)}$. Thus, we must have $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_2 < r$ for all $k > k_0$.

Similar to (55), for all $K > k_0$, we have

$$\sum_{k=k_0}^{K-1} \|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_2 \leqslant \frac{2\sigma}{\mu}\big(\varphi(\Delta_{k_0}) - \varphi(\Delta_K)\big) + (\delta_{k_0} - \delta_K) \leqslant \frac{2\sigma}{\mu}\varphi(\Delta_{k_0}) + \delta_{k_0} < r.$$

Thus the sequence $\{\boldsymbol{x}^{(k)}\}$ is Cauchy and therefore convergent. Since $\boldsymbol{x}^*$ is a limit point of $\{\boldsymbol{x}^{(k)}\}$, it follows that the entire sequence converges to the critical point $\boldsymbol{x}^*$. □

Under additional assumptions, this critical point is in fact guaranteed to be a local minimizer.

Lemma F.1. *Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$ be symmetric and let $\mathbf{A} \succ \mathbf{0}$. If $\|\mathbf{A}^{-1}\mathbf{B}\|_2 < 1$, then $\mathbf{A} - \mathbf{B} \succ \mathbf{0}$.*

Proof. Let $\mathbf{C} = \mathbf{A}^{-1/2}\mathbf{B}\mathbf{A}^{-1/2}$. Since $\mathbf{A}, \mathbf{B}$ are symmetric, so is $\mathbf{C}$. For any $\boldsymbol{x} \neq \mathbf{0}$, setting $\boldsymbol{z} = \mathbf{A}^{1/2}\boldsymbol{x} \neq \mathbf{0}$ gives

$$\boldsymbol{x}^\top(\mathbf{A} - \mathbf{B})\boldsymbol{x} = \|\boldsymbol{z}\|_2^2 - \boldsymbol{z}^\top\mathbf{C}\boldsymbol{z}.$$

Since $\mathbf{C} = \mathbf{A}^{1/2}(\mathbf{A}^{-1}\mathbf{B})\mathbf{A}^{-1/2}$ is similar to $\mathbf{A}^{-1}\mathbf{B}$, their eigenvalues are same and we get

$$\|\mathbf{C}\|_2 = \rho(\mathbf{C}) = \rho(\mathbf{A}^{-1}\mathbf{B}) \leqslant \|\mathbf{A}^{-1}\mathbf{B}\|_2 < 1, \quad (\rho: \text{ spectral radius})$$

The first equality uses symmetry of $\mathbf{C}$. Hence $\boldsymbol{z}^\top\mathbf{C}\boldsymbol{z} \leqslant \|\mathbf{C}\|_2\|\boldsymbol{z}\|_2^2 < \|\boldsymbol{z}\|_2^2$, giving $\boldsymbol{x}^\top(\mathbf{A} - \mathbf{B})\boldsymbol{x} > 0$, hence $\mathbf{A} - \mathbf{B} \succ \mathbf{0}$. □

F.6 Proof of Proposition 4.6

Statement: Let $\boldsymbol{x}^*$ be a critical fixed point of $\mathcal{T}$. Suppose that

$$\|\nabla\mathcal{T}(\boldsymbol{x}^*)\|_2 < 1, \tag{56}$$

and assume that all components of $\boldsymbol{x}^*$ are nonzero. Then $\boldsymbol{x}^*$ is a strict local minimizer of $\mathcal{H}$, and $\mathcal{H}$ satisfies the Kurdyka–Łojasiewicz inequality at $\boldsymbol{x}^*$ with exponent $\theta = 1/2$. Consequently, in the strict DC regime, any DOCH sequence converging to $\boldsymbol{x}^*$ does so linearly.

Proof. To prove that $\boldsymbol{x}^*$ is a strict local minimizer of $\mathcal{H}$, it suffices to show that the Hessian $\nabla^2\mathcal{H}(\boldsymbol{x}^*)$ is positive definite. It follows from definition (32) that

$$\nabla^2\mathcal{H}(\boldsymbol{x}^*) = \nabla^2 f(\boldsymbol{x}^*) - \nabla^2 g(\boldsymbol{x}^*) = 3\beta\,\mathrm{diag}(\boldsymbol{x}^{*2}) - (\mathbf{J} + \alpha\mathbf{I}), \tag{57}$$

where $\boldsymbol{x}^{*2}$ denotes componentwise squaring and $\mathrm{diag}(\boldsymbol{z})$ denotes a diagonal matrix such that $\mathrm{diag}(\boldsymbol{z})_{ii} = z_i$. On the other hand, applying the chain rule of differentiation to (80), we have

$$\nabla\mathcal{T}(\boldsymbol{x}^*) = \frac{1}{3\beta^{1/3}}\mathrm{diag}\big[(\mathbf{A}\boldsymbol{x}^*)^{-2/3}\big]\mathbf{A}, \qquad (\mathbf{A} = \mathbf{J} + \alpha\mathbf{I}). \tag{58}$$

This is where we need the assumption that the components of $\boldsymbol{x}^*$ are nonzero, namely, to ensure that the Jacobian is well-defined at $\boldsymbol{x}^*$. We can simplify (58) using the fact that $\boldsymbol{x}^*$ is a fixed point of $\mathcal{T}$. Indeed, letting $k \to \infty$ in (79), we have

$$\boldsymbol{x}^* = \mathcal{T}(\boldsymbol{x}^*) = \varphi(\beta^{-1}\mathbf{A}\boldsymbol{x}^*),$$

that is, $\mathbf{A}\boldsymbol{x}^* = \beta\,(\boldsymbol{x}^*)^3$. Plugging this in (58), we obtain

$$\nabla\mathcal{T}(\boldsymbol{x}^*) = \frac{1}{3\beta}\,\mathrm{diag}(\boldsymbol{x}^*)^{-2}(\mathbf{J} + \alpha\mathbf{I}). \tag{59}$$

Set $\mathbf{A}_0 = 3\beta\,\mathrm{diag}(\boldsymbol{x}^{*2}) \succ \mathbf{0}$ and $\mathbf{B}_0 = \mathbf{J} + \alpha\mathbf{I}$. Equation (59) gives $\nabla\mathcal{T}(\boldsymbol{x}^*) = \mathbf{A}_0^{-1}\mathbf{B}_0$. Lemma F.1 therefore shows that $\|\nabla\mathcal{T}(\boldsymbol{x}^*)\|_2 < 1$ implies $\mathbf{A}_0 - \mathbf{B}_0 = \nabla^2\mathcal{H}(\boldsymbol{x}^*) \succ \mathbf{0}$. This establishes our first claim.

We next establish that the sharp KL exponent of $\mathcal{H}$ at $\boldsymbol{x}^*$ is $1/2$. The preceding argument shows that

$$\mathbf{H}_* := \nabla^2\mathcal{H}(\boldsymbol{x}^*) = 3\beta\,\mathrm{diag}\big((x_i^*)^2\big) - (\mathbf{J} + \alpha\mathbf{I}) \succ \mathbf{0}.$$

Set

$$m := \frac{1}{2}\lambda_{\min}(\mathbf{H}_*) > 0.$$

Since $\mathcal{H}$ is a polynomial, its Hessian is continuous. Hence there exists $r > 0$ such that

$$\|\nabla^2\mathcal{H}(\boldsymbol{x}) - \mathbf{H}_*\|_2 < m \qquad \text{whenever } \|\boldsymbol{x} - \boldsymbol{x}^*\|_2 < r.$$

Consequently,

$$\nabla^2\mathcal{H}(\boldsymbol{x}) \succeq mI \qquad \text{for every } \boldsymbol{x} \in U := B_r(\boldsymbol{x}^*). \tag{60}$$

Because U is convex, the segment joining any $\boldsymbol{x}, \mathbf{y} \in U$ remains in U. Taylor's formula with integral remainder therefore gives

$$\begin{aligned}\mathcal{H}(\mathbf{y}) - \mathcal{H}(\boldsymbol{x}) - \nabla\mathcal{H}(\boldsymbol{x})^\top(\mathbf{y} - \boldsymbol{x}) &= \int_0^1 (1-t)(\mathbf{y} - \boldsymbol{x})^\top\nabla^2\mathcal{H}\big(\boldsymbol{x} + t(\mathbf{y} - \boldsymbol{x})\big)(\mathbf{y} - \boldsymbol{x})\,dt \\ &\geqslant \frac{m}{2}\|\mathbf{y} - \boldsymbol{x}\|_2^2.\end{aligned}$$

Thus $\mathcal{H}$ is m-strongly convex on U.

Taking the base point to be $\boldsymbol{x}^*$ and using $\nabla\mathcal{H}(\boldsymbol{x}^*) = 0$, we obtain the local quadratic-growth bound

$$\mathcal{H}(\boldsymbol{x}) - \mathcal{H}(\boldsymbol{x}^*) \geqslant \frac{m}{2}\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2, \qquad \boldsymbol{x} \in U. \tag{61}$$

In particular, the objective gap is strictly positive for $\boldsymbol{x} \in U \setminus \{\boldsymbol{x}^*\}$.

Conversely, applying strong convexity with $\mathbf{y} = \boldsymbol{x}^*$ yields

$$\mathcal{H}(\boldsymbol{x}) - \mathcal{H}(\boldsymbol{x}^*) \leqslant \nabla\mathcal{H}(\boldsymbol{x})^\top(\boldsymbol{x} - \boldsymbol{x}^*) - \frac{m}{2}\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2.$$

For any vectors $\mathbf{a}, \mathbf{b}$, completing the square gives

$$\mathbf{a}^\top\mathbf{b} - \frac{m}{2}\|\mathbf{b}\|_2^2 = \frac{\|\mathbf{a}\|_2^2}{2m} - \frac{m}{2}\left\|\mathbf{b} - \frac{\mathbf{a}}{m}\right\|_2^2 \leqslant \frac{\|\mathbf{a}\|_2^2}{2m}.$$

Using $\mathbf{a} = \nabla\mathcal{H}(\boldsymbol{x})$ and $\mathbf{b} = \boldsymbol{x} - \boldsymbol{x}^*$, we conclude that

$$\mathcal{H}(\boldsymbol{x}) - \mathcal{H}(\boldsymbol{x}^*) \leqslant \frac{1}{2m}\|\nabla\mathcal{H}(\boldsymbol{x})\|_2^2.$$

Equivalently,

$$\|\nabla\mathcal{H}(\boldsymbol{x})\|_2 \geqslant \sqrt{2m}\left(\mathcal{H}(\boldsymbol{x}) - \mathcal{H}(\boldsymbol{x}^*)\right)^{1/2}, \qquad \boldsymbol{x} \in U. \tag{62}$$

This is the KL inequality with exponent $\theta = 1/2$.

Also, the exponent $1/2$ is sharp. Fix any unit vector $\mathbf{v}$. Taylor expansion at $\boldsymbol{x}^*$ gives

$$\mathcal{H}(\boldsymbol{x}^* + t\mathbf{v}) - \mathcal{H}(\boldsymbol{x}^*) = \frac{t^2}{2}\mathbf{v}^\top\mathbf{H}_*\mathbf{v} + O(|t|^3),$$

and

$$\nabla\mathcal{H}(\boldsymbol{x}^* + t\mathbf{v}) = t\mathbf{H}_*\mathbf{v} + O(t^2).$$

Since $\mathbf{H}_* \succ 0$, both $\mathbf{v}^\top\mathbf{H}_*\mathbf{v}$ and $\|\mathbf{H}_*\mathbf{v}\|_2$ are positive. Therefore, for every $\theta < 1/2$,

$$\frac{\|\nabla\mathcal{H}(\boldsymbol{x}^* + t\mathbf{v})\|_2}{\left(\mathcal{H}(\boldsymbol{x}^* + t\mathbf{v}) - \mathcal{H}(\boldsymbol{x}^*)\right)^\theta} = \frac{\|\mathbf{H}_*\mathbf{v}\|_2}{\left(\frac{1}{2}\mathbf{v}^\top\mathbf{H}_*\mathbf{v}\right)^\theta}|t|^{1-2\theta}(1 + o(1)) \longrightarrow 0.$$

Thus no positive constant can make the KL inequality hold locally with exponent $\theta < 1/2$. The sharp KL exponent at $\boldsymbol{x}^*$ is thus $1/2$.

Finally, for the convergence-rate assertion, suppose that $\alpha > \lambda_{\max}(-\mathbf{J})$, $\beta > 0$, and $\boldsymbol{x}^{(k)} \to \boldsymbol{x}^*$. Propositions 4.1 and 4.3 provide the sufficient-decrease and relative-error estimates

$$\mathcal{H}(\boldsymbol{x}^{(k)}) - \mathcal{H}(\boldsymbol{x}^{(k+1)}) \geqslant \frac{\mu}{2}\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_2^2,$$
$$\|\nabla\mathcal{H}(\boldsymbol{x}^{(k+1)})\|_2 \leqslant \sigma\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_2,$$

where $\mu = \lambda_{\min}(\mathbf{J} + \alpha\mathbf{I}) > 0$ and $\sigma > 0$. Together with continuity of $\mathcal{H}$ and the KL exponent $\theta = 1/2$ at $\boldsymbol{x}^*$, these estimates satisfy the hypotheses of the KL convergence-rate theorem Frankel et al. (2015). Hence the iterates converge linearly to $\boldsymbol{x}^*$. □

F.7 Proof of Proposition 4.7

Statement. Suppose $\mathbf{A} \succeq \mathbf{0}$, $\beta > 0$, and $q < \infty$. Let $\mu = \lambda_{\min}(\mathbf{A}) \geqslant 0$. Then the ADOCH iterates are bounded, $\{M_k\}$ is non-increasing, and

$$\mathcal{H}(\boldsymbol{x}^{(k+1)}) \leqslant M_k - \frac{\mu}{2}\|\boldsymbol{d}_k\|_2^2 - \frac{\beta}{12n}\|\boldsymbol{d}_k\|_2^4. \tag{63}$$

Moreover,

$$\lim_{k\to\infty}\min_{k\leqslant j\leqslant k+q}\|\nabla\mathcal{H}(\boldsymbol{x}^{(j+1)})\|_2 = 0. \tag{64}$$

If $\mathbf{A} \succ \mathbf{0}$, then, with $\mathcal{H}_{\inf} = \inf_{\boldsymbol{x}\in\mathbb{R}^n}\mathcal{H}(\boldsymbol{x})$, for every $N \geqslant 0$,

$$\min_{0\leqslant j\leqslant N+q}\|\nabla\mathcal{H}(\boldsymbol{x}^{(j+1)})\|_2 \leqslant \|\mathbf{A}\|_2\sqrt{\frac{2(q+1)(\mathcal{H}(\boldsymbol{x}^{(0)}) - \mathcal{H}_{\inf})}{\mu(N+1)}}. \tag{65}$$

We first record two identities that use the particular quartic structure of the Hamiltonian. Write

$$\mathbf{A} = \mathbf{J} + \alpha\mathbf{I}, \qquad D_f(\boldsymbol{u}, \boldsymbol{z}) = f(\boldsymbol{u}) - f(\boldsymbol{z}) - \nabla f(\boldsymbol{z})^\top(\boldsymbol{u} - \boldsymbol{z}).$$

At iteration k, let $\boldsymbol{u} = \boldsymbol{x}^{(k+1)}$, $\boldsymbol{z} = \boldsymbol{v}^{(k)}$, and $\boldsymbol{d}_k = \boldsymbol{u} - \boldsymbol{z}$. The update $\boldsymbol{u} = \mathcal{T}(\boldsymbol{z})$ is equivalent to

$$\nabla f(\boldsymbol{u}) = \mathbf{A}\boldsymbol{z}. \tag{66}$$

Consequently,

$$\nabla\mathcal{H}(\boldsymbol{x}^{(k+1)}) = \mathbf{A}(\boldsymbol{v}^{(k)} - \boldsymbol{x}^{(k+1)}) = -\mathbf{A}\boldsymbol{d}_k. \tag{67}$$

Since $g(\boldsymbol{x}) = \boldsymbol{x}^\top\mathbf{A}\boldsymbol{x}/2$ is quadratic, direct expansion using (66) gives the exact energy identity

$$\mathcal{H}(\boldsymbol{v}^{(k)}) - \mathcal{H}(\boldsymbol{x}^{(k+1)}) = D_f(\boldsymbol{v}^{(k)}, \boldsymbol{x}^{(k+1)}) + \frac{1}{2}\boldsymbol{d}_k^\top\mathbf{A}\boldsymbol{d}_k. \tag{68}$$

For the scalar function $r(t) = \beta t^4/4$, put $h = a - b$. Completing the square yields

$$D_r(a,b) = \beta\left(\frac{3}{2}b^2h^2 + bh^3 + \frac{1}{4}h^4\right)$$
$$= \frac{3\beta}{2}h^2\left(b + \frac{h}{3}\right)^2 + \frac{\beta}{12}h^4 \geqslant \frac{\beta}{12}|a-b|^4. \tag{69}$$

Summing (69) over the coordinates and using $\|\boldsymbol{z}\|_4^4 \geqslant n^{-1}\|\boldsymbol{z}\|_2^4$ gives

$$D_f(\boldsymbol{u},\boldsymbol{z}) \geqslant \frac{\beta}{12}\|\boldsymbol{u}-\boldsymbol{z}\|_4^4 \geqslant \frac{\beta}{12n}\|\boldsymbol{u}-\boldsymbol{z}\|_2^4. \tag{70}$$

We now prove the proposition. By the acceptance test (8), $\mathcal{H}(\boldsymbol{v}^{(k)}) \leqslant M_k$ whether the extrapolated point is accepted or rejected. If $\mathbf{A} \succeq \mathbf{0}$, then (68)–(70) imply

$$\mathcal{H}(\boldsymbol{x}^{(k+1)}) \leqslant M_k - \Psi_k, \qquad \Psi_k = \frac{\mu}{2}\|\boldsymbol{d}_k\|_2^2 + \frac{\beta}{12n}\|\boldsymbol{d}_k\|_2^4. \tag{71}$$

This is (14). Every old value entering M_{k+1} is at most M_k, and the new value is at most M_k by (71); hence $M_{k+1} \leqslant M_k$. In particular, $\mathcal{H}(\boldsymbol{x}^{(k)}) \leqslant M_0 = \mathcal{H}(\boldsymbol{x}^{(0)})$ for every k. Coercivity of $\mathcal{H}$ therefore proves boundedness of the iterates.

For any $k \geqslant 0$, monotonicity of M_k and (71) give

$$M_k - M_{k+q+1} \geqslant \min_{k \leqslant j \leqslant k+q} \Psi_j. \tag{72}$$

Indeed, M_{k+q+1} is the maximum of the $q+1$ values $\mathcal{H}(\boldsymbol{x}^{(j+1)})$, $j = k, \dots, k+q$, and each of them is at most $M_k - \Psi_j$. The non-increasing sequence $\{M_k\}$ is bounded below by $\mathcal{H}_{\inf}$ and thus converges. The left-hand side of (72) tends to zero. Since $\Psi_j \geqslant \beta\|\boldsymbol{d}_j\|_2^4/(12n)$, it follows that

$$\lim_{k\to\infty} \min_{k \leqslant j \leqslant k+q} \|\boldsymbol{d}_j\|_2 = 0.$$

Combining this with the exact gradient identity (67) proves (15). In particular, ADOCH always has a subsequence converging to the critical set in the positive-semidefinite regime.

It remains to prove the rate. Sum (72) for $k = 0, \dots, N$. Splitting the left-hand sum into the $q+1$ residue classes modulo $q+1$ produces $q+1$ telescoping chains, so

$$(N+1)\min_{0 \leqslant j \leqslant N+q} \Psi_j \leqslant \sum_{k=0}^{N}(M_k - M_{k+q+1}) \leqslant (q+1)(\mathcal{H}(\boldsymbol{x}^{(0)}) - \mathcal{H}_{\inf}). \tag{73}$$

If $\mu > 0$, use $\Psi_j \geqslant \mu\|\boldsymbol{d}_j\|_2^2/2$ and (67) in (73) to obtain (16). If $\mu = 0$, the quartic term instead gives the valid semidefinite bound

$$\min_{0 \leqslant j \leqslant N+q} \|\nabla\mathcal{H}(\boldsymbol{x}^{(j+1)})\|_2 \leqslant \|\mathbf{A}\|_2\left(\frac{12n(q+1)(\mathcal{H}(\boldsymbol{x}^{(0)}) - \mathcal{H}_{\inf})}{\beta(N+1)}\right)^{1/4}. \tag{74}$$

This completes the proof.

F.8 Proof of Theorem 4.8

Statement. Suppose $\mathbf{A} \succeq \mathbf{0}$ and $\beta > 0$. If ADOCH uses $q = 0$ from the beginning, or switches permanently to $q = 0$ after any finite iteration, then $\{\boldsymbol{x}^{(k)}\}$ converges to a critical point of $\mathcal{H}$.

Discarding a finite prefix does not affect convergence, so suppose without loss of generality that $q = 0$ throughout. Then $M_k = \mathcal{H}(\boldsymbol{x}^{(k)})$, and summing (71) gives

$$\sum_{k=0}^{\infty}\|\boldsymbol{d}_k\|_2^4 \leqslant \frac{12n}{\beta}\big(\mathcal{H}(\boldsymbol{x}^{(0)}) - \mathcal{H}_{\inf}\big) < \infty. \tag{75}$$

Hence $\boldsymbol{d}_k \to \mathbf{0}$, and (67) implies $\nabla\mathcal{H}(\boldsymbol{x}^{(k)}) \to \mathbf{0}$. The iterates are bounded by Proposition 4.7.

We next use a property specific to the quartic Hamiltonian: its critical set is finite. Indeed, over $\mathbb{C}$, the critical equations are

$$p_i(\boldsymbol{x}) = \beta x_i^3 - \sum_{j=1}^{n} A_{ij} x_j = 0, \qquad i = 1, \dots, n. \tag{76}$$

Under any degree-compatible monomial order, the leading monomials of $p_1, \dots, p_n$ are $x_1^3, \dots, x_n^3$. They are pairwise relatively prime, so Buchberger's criterion shows that these polynomials form a Gröbner basis Cox et al. (2015). The residue classes of the monomials $x_1^{a_1} \cdots x_n^{a_n}$ with $0 \leqslant a_i \leqslant 2$ therefore form a basis of $\mathbb{C}[x_1, \dots, x_n]/\langle p_1, \dots, p_n \rangle$ Cox et al. (2015). This quotient has dimension 3^n, so (76) is zero-dimensional and has at most 3^n distinct complex solutions, counting multiplicity. In particular, the real critical set

$$\mathcal{C} = \{\boldsymbol{x} \in \mathbb{R}^n : \nabla \mathcal{H}(\boldsymbol{x}) = \mathbf{0}\}$$

is finite.

Boundedness and $\nabla \mathcal{H}(\boldsymbol{x}^{(k)}) \to \mathbf{0}$ imply

$$\operatorname{dist}(\boldsymbol{x}^{(k)}, \mathcal{C}) \longrightarrow 0. \tag{77}$$

Otherwise a bounded subsequence staying a positive distance from $\mathcal{C}$ would have a limit point outside $\mathcal{C}$ at which the continuous gradient vanishes, a contradiction. For each k, choose a nearest point $\boldsymbol{c}_k \in \mathcal{C}$ to $\boldsymbol{x}^{(k)}$. By (77), $\|\boldsymbol{x}^{(k)} - \boldsymbol{c}_k\|_2 \to 0$.

Let $\gamma_k = (t_k - 1)/t_{k+1}$. The Nesterov recurrence gives $0 \leqslant \gamma_k < 1$ and $\gamma_k \to 1$. If the extrapolated point is rejected at iteration k, then

$$\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)} = \boldsymbol{d}_k \longrightarrow \mathbf{0},$$

so $\boldsymbol{c}_{k+1} = \boldsymbol{c}_k$ for all sufficiently large rejected iterations, because $\mathcal{C}$ is finite. If the extrapolated point is accepted, then

$$\boldsymbol{x}^{(k+1)} - (1 + \gamma_k)\boldsymbol{x}^{(k)} + \gamma_k \boldsymbol{x}^{(k-1)} = \boldsymbol{d}_k \longrightarrow \mathbf{0}.$$

Using (77), boundedness of $\mathcal{C}$, and $\gamma_k \to 1$, we obtain

$$\boldsymbol{c}_{k+1} - 2\boldsymbol{c}_k + \boldsymbol{c}_{k-1} \longrightarrow \mathbf{0}$$

along accepted iterations. The displayed second differences take values in a finite set, and hence they are exactly zero at every sufficiently large accepted iteration.

Define $\boldsymbol{\Delta}_k = \boldsymbol{c}_k - \boldsymbol{c}_{k-1}$. On the tail, a rejected step gives $\boldsymbol{\Delta}_{k+1} = \mathbf{0}$, while an accepted step gives $\boldsymbol{\Delta}_{k+1} = \boldsymbol{\Delta}_k$. If a rejection occurs, all subsequent increments consequently remain zero. If no rejection occurs, all subsequent increments are equal; boundedness of the finite-valued sequence $\{\boldsymbol{c}_k\}$ forces this common increment to be zero. Thus $\boldsymbol{c}_k$ is eventually constant. Equation (77) now shows that $\boldsymbol{x}^{(k)}$ converges to that critical point, proving the theorem.

F.9 Boundedness outside the DC regime and necessity of convexity

The accelerated iterates remain bounded even when $\mathbf{A}$ is indefinite, although convergence and stationarity can then fail.

Proposition F.2. *For every symmetric $\mathbf{A} = \mathbf{J} + \alpha\mathbf{I}$, every $\beta > 0$, every finite q, and every initialization, the ADOCH iterates are bounded.*

Proof. Let $c = \beta^{-1}\|\mathbf{A}\|_\infty$ and

$$R = \max\{\|\boldsymbol{x}^{(0)}\|_\infty, \sqrt{3c}\}.$$

The case $R = 0$ is immediate. Since $0 \leqslant \gamma_k < 1$, if $\|\boldsymbol{x}^{(k-1)}\|_\infty$ and $\|\boldsymbol{x}^{(k)}\|_\infty$ are at most R, then

$$\|\boldsymbol{v}^{(k)}\|_\infty \leqslant \max\{\|\boldsymbol{x}^{(k)}\|_\infty, \|\boldsymbol{y}^{(k)}\|_\infty\} \leqslant 3R.$$

The componentwise cube-root update therefore satisfies

$$\|\boldsymbol{x}^{(k+1)}\|_\infty \leqslant (c\|\boldsymbol{v}^{(k)}\|_\infty)^{1/3} \leqslant (3cR)^{1/3} \leqslant R.$$

The same estimate applies at $k = 0$ because $\boldsymbol{y}^{(0)} = \boldsymbol{x}^{(0)}$. Induction proves the claim. □

We have already seen that for sufficiently large α, g is convex (Proposition 3.1), and boundedness follows directly from the DCP structure. When α is small, $\mathcal{H}$ is no longer a difference of convex functions. However, boundedness can still be ensured by an appropriate choice of β.

Proposition F.3. *The DOCH iterates $\{\boldsymbol{x}^{(k)}\}$ are bounded if $\alpha > 0$ and*

$$\beta \geqslant \|\mathbf{J} + \alpha\mathbf{I}\|_\infty = \max_{1\leqslant i\leqslant n} \Big(\alpha + \sum_{j\neq i} |J_{ij}|\Big). \tag{78}$$

Proof. Starting with an initial point $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$, the DOCH iterates are generated using

$$\boldsymbol{x}^{(k+1)} = \mathcal{T}(\boldsymbol{x}^{(k)}), \tag{79}$$

where

$$\mathcal{T}(\boldsymbol{x}) = \varphi\big(\beta^{-1}(\mathbf{J} + \alpha\mathbf{I})\boldsymbol{x}\big) \qquad \text{and} \qquad \varphi(x_1, \ldots, x_n) = (\sqrt[3]{x}_1, \ldots, \sqrt[3]{x}_n). \tag{80}$$

The matrix norm $\|\cdot\|_\infty$ is induced by the vector norm $\|\boldsymbol{z}\|_\infty = \max_{1\leqslant i\leqslant n} |z_i|$. Hence,

$$\|\beta^{-1}(\mathbf{J} + \alpha\mathbf{I})\boldsymbol{x}\|_\infty \leqslant \beta^{-1}\|\mathbf{J} + \alpha\mathbf{I}\|_\infty \|\boldsymbol{x}\|_\infty.$$

On the other hand,

$$\|\varphi(\boldsymbol{z})\|_\infty = \|\boldsymbol{z}\|_\infty^{1/3}.$$

Combining these, we have from (80) that

$$\|\mathcal{T}(\boldsymbol{x})\|_\infty \leqslant \big(\beta^{-1}\|\mathbf{J} + \alpha\mathbf{I}\|_\infty\big)^{1/3} \|\boldsymbol{x}\|_\infty^{1/3}.$$

In particular, $\|\mathcal{T}(\boldsymbol{x})\|_\infty \leqslant \|\boldsymbol{x}\|_\infty^{1/3}$ if $\beta \geqslant \|\mathbf{J} + \alpha\mathbf{I}\|_\infty$. Thus, letting $\boldsymbol{x} = \boldsymbol{x}^{(k)}$, we get

$$\|\boldsymbol{x}^{(k+1)}\|_\infty = \|\mathcal{T}(\boldsymbol{x}^{(k)})\|_\infty \leqslant \|\boldsymbol{x}^{(k)}\|_\infty^{1/3}.$$

From this estimate, we can easily verify using induction that $\|\boldsymbol{x}^{(k)}\|_\infty \leqslant \max(1, \|\boldsymbol{x}^{(0)}\|_\infty)$ for all $k \geqslant 1$. This shows that the iterates are bounded. □

G Ablation Study & Parameter Settings

Remark G.1. In the strict DC regime, Theorem 4.5 guarantees that DOCH converges to a critical point. Proposition 4.7 gives the corresponding finite-window guarantee for ADOCH, and Theorem 4.8 gives full convergence after a permanent $q = 0$ switch. If the limiting fixed point has no zero component and also satisfies (56), Proposition 4.6 further makes it a strict local minimizer of $\mathcal{H}$, and the DOCH iterates converges linearly to that point. The experiments in Figure 3a,c illustrate these conditions. The scale $\beta = n\sqrt{n}\|\mathbf{J} + \alpha\mathbf{I}\|_\infty$ satisfies the sufficient boundedness condition in Proposition F.3; the additional factor $n\sqrt{n}$ was selected empirically and also produced favourable energies in Figure 3b.

Remark G.2. In practice, the fixed points of $\mathcal{T}$ have nonzero components, which are rounded to get the spin vector. If these fixed points are stable in the sense of (56), then Proposition 4.6 ensures that the iterates $\{\boldsymbol{x}^{(k)}\}$ converge linearly to a local minimum of the Hamiltonian (see Figure 3a). Empirically, we found that setting β sufficiently large, on the order of $\mathcal{O}\left(n\sqrt{n}\|\mathbf{J} + \alpha\mathbf{I}\|_\infty\right)$, as stipulated by Proposition F.3, keeps the Jacobian norm at the fixed point below one (see Figure 3c). We found that this also yields better quality solutions (see Figure 3b). In summary, this means that we can set β to a high value and tune α freely.

Remark G.3. In the strict DC regime, the DOCH convergence result motivates monitoring the relative change

$$r^{(k)} = \frac{\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_2}{\|\boldsymbol{x}^{(k)}\|_2},$$

and stopping once $r^{(k)} < \epsilon$, for some tolerance $\epsilon > 0$. On the K_{2000} graph, this rule detected the observed saturation point without degrading the final energy (Figure 3d). For ADOCH, the theoretically justified residual is instead

$$r_{\mathrm{A}}^{(k)} = \frac{\|\boldsymbol{x}^{(k+1)} - \boldsymbol{v}^{(k)}\|_2}{\max\{1, \|\boldsymbol{v}^{(k)}\|_2\}}.$$

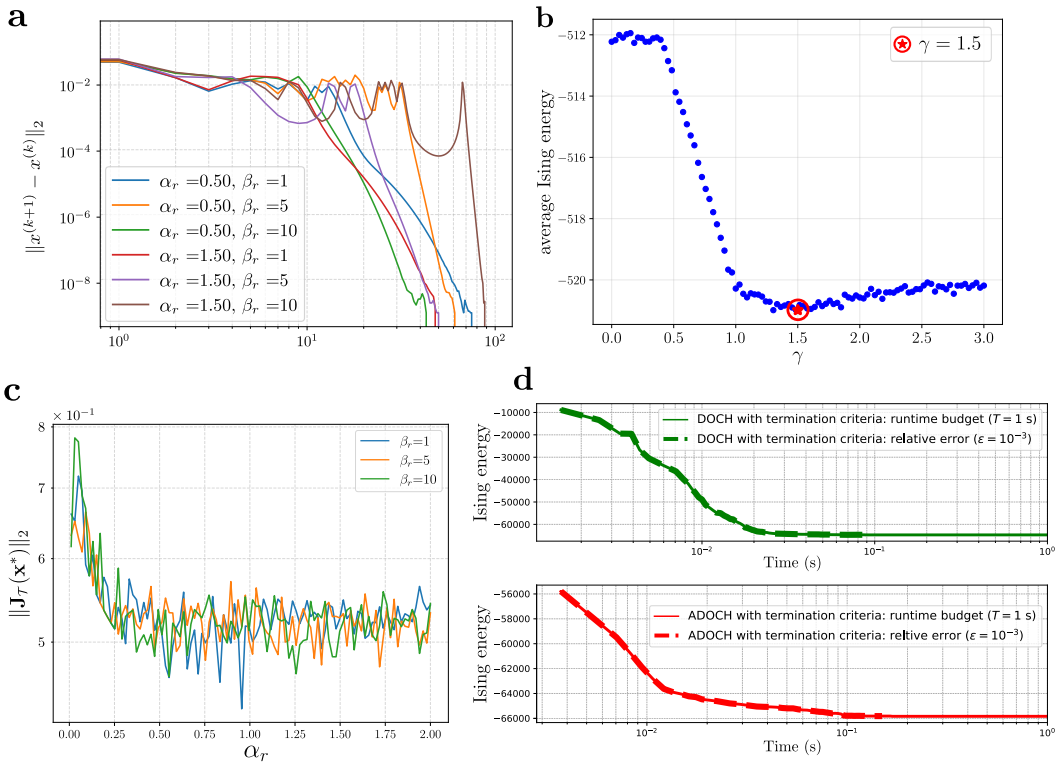


Figure 3: **Ablation study of the DOCH solver on 100-spin SK and K_{2000} Ising models. (a)** Log-log convergence behaviour of the DOCH fixed-point iteration (79), showing the norm of successive iterate differences for a 100-spin SK model. Rapid convergence toward a fixed point $\boldsymbol{x}^*$ is observed within 100 iterations for different hyperparameter ratios $\alpha_r \in \{0.50, 1.50\}$ and $\beta_r \in \{1, 5, 10\}$. **(b)** Average Ising energy versus exponent γ in $\beta = n^\gamma \|\mathbf{J} + \alpha\mathbf{I}\|_\infty$ for a 100-spin SK model. For each $\gamma \in (0, 3]$, DOCH is run for 100 iterations with optimal α and multiple random initializations $\boldsymbol{x}^{(0)}$. The lowest average energy occurs near $\gamma = 1.5$. **(c)** Dependence of the average Jacobian spectral norm $\|\nabla\mathcal{T}(\boldsymbol{x}^*)\|_2$ on the ratio $\alpha_r = \alpha/\lambda_{\max}(-\mathbf{J})$ for $\beta_r \in \{1, 5, 10\}$ in a 100-spin SK model. Fixed points are obtained after 100 iterations and the Jacobian norm is averaged over multiple random initializations. **(d)** Comparison of termination criteria for DOCH and ADOCH on the K_{2000} graph: a fixed runtime budget ($T = 1$ s) versus a successive-iterate relative tolerance ($\epsilon = 10^{-3}$). For ADOCH this displayed tolerance is an empirical proxy; the accepted-point residual $\|\boldsymbol{x}^{(k+1)} - \boldsymbol{v}^{(k)}\|_2$ is the stationarity certificate. All experiments were performed on a laptop with a 4GB NVIDIA RTX 3050 GPU.

Indeed, (67) implies

$$\mu\|\boldsymbol{x}^{(k+1)} - \boldsymbol{v}^{(k)}\|_2 \leqslant \|\nabla\mathcal{H}(\boldsymbol{x}^{(k+1)})\|_2 \leqslant \|\mathbf{J} + \alpha\mathbf{I}\|_2\|\boldsymbol{x}^{(k+1)} - \boldsymbol{v}^{(k)}\|_2.$$

The successive-iterate ADOCH rule displayed in Figure 3d is therefore an empirical proxy for this certificate.

Our solvers have two Hamiltonian parameters, α and β. Define $\lambda_- = \lambda_{\max}(-\mathbf{J}) = -\lambda_{\min}(\mathbf{J}) \geqslant 0$ and let $\widehat{\lambda}_-$ denote the value used in computation. We use

$$\alpha = \eta\widehat{\lambda}_- \quad \text{and} \quad \beta = n\sqrt{n}\,\|\mathbf{J} + \alpha\mathbf{I}\|_\infty = n\sqrt{n}\,\max_{1\leqslant i\leqslant n}\left(\alpha + \sum_{j\neq i}|J_{ij}|\right), \tag{81}$$

where η is tuned. If $\widehat{\lambda}_- = \lambda_-$ exactly, then $\eta > 1$ gives the strict DC regime, $\eta = 1$ makes g convex but not necessarily strongly convex, and $\eta < 1$ lies outside the convergence regimes of Theorems 4.5 and 4.8. The formula for β is stronger than the sufficient condition in Proposition F.3, and therefore bounds the DOCH iterates for every $\alpha > 0$; Proposition F.2 gives boundedness of ADOCH for every $\beta > 0$.

For the small and moderate instances, λ_- is computed directly or by the power method applied to $-\mathbf{J}$ Horn & Johnson (1985). For the large instances, we use the Wigner-inspired estimate Mehta (2004); Goto (2019); Goto et al. (2021)

$$\widehat{\lambda}_- = 2\sigma_J\sqrt{n}, \qquad \sigma_J^2 = \frac{1}{n(n-1)}\sum_{i\neq j}(J_{ij}-\bar{J})^2, \qquad \bar{J} = \frac{1}{n(n-1)}\sum_{i\neq j}J_{ij}. \tag{82}$$

This estimate is a parameter heuristic, not an eigenvalue certificate. So, as per Remark G.2, we tune η over $(0, 2]$ using the common adaptive box search described in Appendix M.

The ADOCH look-back parameter is $q = 2$ in the headline ultra-large experiments (Figures 6c,d and 16). In other experiments, $q \leqslant 5$ for $n \geqslant 10^4$ and was selected from $\{2, \dots, 10\}$ for smaller models. These choices use Proposition 4.7; a permanent switch to $q = 0$ is the convergence-safe termination mode of Theorem 4.8.

H MAX-CUT AND GW-SDP

We explain how the MAX-CUT problem on a graph can be reformulated as an Ising model and how the Goemans–Williamson Semidefinite Program (GW-SDP) Goemans & Williamson (1995) can be used to approximate its ground state.

Given a simple undirected graph $G = (V, E)$, the MAX-CUT problem seeks a partition of the vertex set V into two disjoint subsets S and $T = V \setminus S$ such that the total weight of edges (called the cut value) across S and T is maximized. Specifically, if ω_{ij} denotes the weight of an edge $(i, j) \in E$, then the cut value is

$$\sum_{\substack{(i,j)\in E\\ i\in S,\, j\in T}} \omega_{ij}. \tag{83}$$

The MAX-CUT problem is to maximize (83) with respect to the choice of subsets S and T.

To reformulate this as an Ising problem, we define the weighted adjacency matrix $\mathbf{W} = \{W_{ij}\}$ given by

$$W_{ij} = \begin{cases} \omega_{ij}, & \text{if } (i,j) \in E, \\ 0, & \text{otherwise.} \end{cases}$$

Note that $W_{ii} = 0$ (no self-loops), and $W_{ij} = W_{ji}$ since the graph is undirected. For any given S and T, we can uniquely associate a spin vector $\boldsymbol{s} \in \{-1, 1\}^n$ such that $s_i = 1$ if $i \in S$, and $s_i = -1$ if $i \in T$. With this encoding, we can write cut value (83) as

$$\frac{1}{4}\sum_{i,j=1}^{n}(1 - s_i s_j)\, W_{ij} = \text{constant} - \frac{1}{4}\sum_{i,j=1}^{n} s_i s_j W_{ij}.$$

Consequently, the MAX-CUT problem becomes:

$$\max_{\boldsymbol{s}\in\{-1,1\}^n} \text{constant} - \frac{1}{4}\sum_{i,j=1}^{n} s_i s_j W_{ij} \equiv \min_{\boldsymbol{s}\in\{-1,1\}^n} -\frac{1}{2}\,\boldsymbol{s}^\top \mathbf{J}\boldsymbol{s}.$$

where $\mathbf{J} = -(1/2)\mathbf{W}$. This is precisely the Ising problem.

The GW-SDP is based on the observation that by introducing the matrix variable $\mathbf{X} = \boldsymbol{s}\boldsymbol{s}^\top$, we can write the Ising energy as

$$\mathcal{E}(\boldsymbol{s}) = -\frac{1}{2}\boldsymbol{s}^\top\mathbf{J}\boldsymbol{s} = -\frac{1}{2}\text{Trace}(\mathbf{J}\boldsymbol{s}\boldsymbol{s}^\top) = -\frac{1}{2}\text{Trace}(\mathbf{J}\mathbf{X}).$$

The variable $\mathbf{X}$ is a rank-one, positive semidefinite matrix ($\mathbf{X} \succeq \mathbf{0}$) with $X_{ii} = 1$ for $i = 1, \dots, n$. There exists a one-to-one correspondence between spin assignments in $\{-1, 1\}^n$ and matrices satisfying these properties. Consequently, the Ising problem can be reformulated as:

$$\min_{\substack{\text{rank}(\mathbf{X})=1\\ \mathbf{X}\succeq\mathbf{0}\\ \mathbf{X}_{ii}=1}} -\frac{1}{2}\text{Trace}(\mathbf{J}\mathbf{X})$$

By dropping the nonconvex rank constraint, we obtain a convex optimization problem known as a semidefinite program:

$$\max_{\substack{\mathbf{X}\succeq\mathbf{0}\\ \mathbf{X}_{ii}=1}} \quad \mathrm{Trace}(\mathbf{J}\,\mathbf{X}) \tag{84}$$

We can solve (84) in polynomial time using standard interior-point methods. However, the computational complexity for achieving a given accuracy is relatively high, namely $\mathcal{O}(n^{4.5})$ for an n-spin Ising model Luo et al. (2010). In practice, this limits scalability to about 10^3 spins. For MAX-CUT with nonnegative edge weights, Goemans and Williamson showed that randomized hyperplane rounding achieves an expected cut value of at least 0.878 times the optimum Goemans & Williamson (1995). This guarantee does not cover signed edge weights, arbitrary Ising interactions, or deterministic sign rounding.

I Existing Ising Solvers

I.1 Simulated Annealing (SA)

We implemented the Simulated Annealing (SA) algorithm following the approach in Inagaki et al. (2016). The pseudocode is provided in Algorithm 3. We used a logarithmic cooling schedule $\beta(t) = \beta_0 \log(1 + t/T)$, where β_0 is the initial inverse temperature parameter and T is the total number of iterations (or runtime). SA has two tunable parameters, β_0 and T. For the K_{2000} Ising model, the ratio $T/\beta_0 \approx 1000$ is recommended in Inagaki et al. (2016); accordingly, we use $\beta_0 = 1$ and $T = 1000$. For other cases, β_0 is selected from the interval $[1, 2]$ to obtain the best performance.

Algorithm 3 Simulated Annealing (SA)

Input: initial spin state $\boldsymbol{s}$, maximum iterations T
$n \leftarrow \dim(\mathbf{J})$
$E \leftarrow \mathcal{E}(\boldsymbol{s})$
for $t = 0$ **to** T **do**
  $\beta \leftarrow \beta_0 \log(1 + t/T)$ ▷ logarithmic cooling schedule
  Randomly pick spin $i \in \{1, 2, \ldots, n\}$
  $\boldsymbol{s}' \leftarrow \mathrm{flip}(\boldsymbol{s}, i)$ ▷ flip the i-th spin: $s_i \leftarrow -s_i$
  $\Delta E \leftarrow \mathcal{E}(\boldsymbol{s}') - E$
  Sample $z \sim \mathrm{Unif}(0, 1)$
  if $\Delta E < 0$ **or** $\exp(-\beta \Delta E) \geqslant z$ **then**
    $\boldsymbol{s} \leftarrow \boldsymbol{s}'$
    $E \leftarrow E + \Delta E$
  end if
end for
Output: $\boldsymbol{s}$

I.2 ballistic Bifurcation Machine (bSB)

We implemented the ballistic Bifurcation Machine following the approach described in Goto et al. (2021). The pseudocode is provided in Algorithm 4. Following the setup in Goto et al. (2021), the pump rate is set to $a_0 = 1$, so that a_t increases linearly from 0 to 1 according to the prescribed pump rate schedule. The time step Δ_t is selected from the set {0.25, 0.5, 0.75, 1, 1.25}, based on empirical performance. The coupling strength c_0 is set using the formula:

$$c_0 = \frac{1}{2\langle \mathbf{J} \rangle \sqrt{n}}, \tag{85}$$

where $\langle \mathbf{J} \rangle$ denotes the sample variance of the entries of the matrix $\mathbf{J}$. For instance, for the K_{2000} model with $J_{ij} = \pm 1$ and $n = 2000$, this yields $c_0 \approx 0.0112$. For other problem instances (e.g., SK, G-Set, Biq Mac), we compute c_0 explicitly based on the corresponding $\mathbf{J}$ matrix for each graph.

I.3 Simulated Coherent Ising Machine (SimCIM)

We implemented the Simulated Coherent Ising Machine following Goto et al. (2021). The pseudocode is provided in Algorithm 5, where $\mathcal{N}[\mathbf{0}, \mathbf{I}]$ denotes the standard n-dimensional Gaussian distribution.

Algorithm 4 Ballistic Simulated Bifurcation Machine (bSB)

Input: pump rate a_0, coupling strength c_0, time step Δ_t, total steps T
$n \leftarrow \dim(\mathbf{J})$
Initialize: $\boldsymbol{x} \leftarrow 2 \cdot \text{Bernoulli}(0.5, n) - 1$ ▷ random ± 1 vector
Initialize: $\boldsymbol{y} \leftarrow \mathbf{0}_n$ ▷ zero momentum vector
for $t = 0$ **to** T **do**
  $a_t \leftarrow (a_0 t/T)$ ▷ pump schedule
  $\boldsymbol{y} \leftarrow \boldsymbol{y} + \big(-(a_0 - a_t)\boldsymbol{x} + c_0 \mathbf{J}\boldsymbol{x} \big)\Delta_t$ ▷ momentum update
  $\boldsymbol{x} \leftarrow \boldsymbol{x} + a_0 \boldsymbol{y} \Delta_t$ ▷ position update
  $\boldsymbol{x} \leftarrow \text{clip}(\boldsymbol{x}, -1, 1)$ ▷ enforce bounds
  $y_i \leftarrow 0$ **if** $x_i = \pm 1$ ▷ zero momentum at boundaries
end for
Output: $\boldsymbol{s} = \text{sign}(\boldsymbol{x})$

We adopt the same parameter settings from Goto et al. (2021) for the pump rate a_0, the coupling strength c_0, and the time step Δ_t. The noise amplitude A is chosen from the interval $[0.1, 1]$, with values in this range empirically yielding the best performance.

Algorithm 5 Simulated Coherent Ising Machine (SimCIM)

Input: noise amplitude A, initial rate a_0, coupling c_0, time step Δ_t, total steps T
$n \leftarrow \dim(\mathbf{J})$
Initialize: $\boldsymbol{x} \leftarrow 2 \cdot \text{Bernoulli}(0.5, n) - 1$ ▷ random ± 1 vector
for $t = 0$ **to** T **do**
  $a_t \leftarrow (a_0 t/T)$ ▷ pump schedule
  $\boldsymbol{w} \sim \mathcal{N}(0, \mathbf{I})$ ▷ Gaussian noise
  $\boldsymbol{x} \leftarrow \boldsymbol{x} + \Big(-(a_0 - a_t)\boldsymbol{x} + c_0 \mathbf{J}\,\text{sign}(\boldsymbol{x}) \Big)\Delta_t + A\,\boldsymbol{w}\sqrt{\Delta_t}$
  $\boldsymbol{x} \leftarrow \text{clip}(\boldsymbol{x}, -1, 1)$ ▷ enforce bounds
end for
Output: $\boldsymbol{s} = \text{sign}(\boldsymbol{x})$

The original Coherent Ising Machine (CIM) Inagaki et al. (2016) is based on a degenerate optical parametric oscillator (DOPO) and is modeled by the following system of stochastic differential equations Wang et al. (2013); Inagaki et al. (2016):

$$dc_i = \left[(-1 + p - c_i^2 - s_i^2)c_i + r \sum_j J_{ij} \tilde{c}_j \right] dt + \frac{1}{A_s} \sqrt{c_i^2 + s_i^2 + 0.5}\, dW_1, \tag{86}$$

$$ds_i = \left[(-1 - p - c_i^2 - s_i^2)s_i \right] dt + \frac{1}{A_s} \sqrt{c_i^2 + s_i^2 + 0.5}\, dW_2, \tag{87}$$

where

- c_i and s_i are the in-phase and quadrature amplitudes of the i-th DOPO,
- $\tilde{c}_i$ is the measured in-phase amplitude (perturbed by measurement noise),
- $\{J_{ij}\}$ are the coupling coefficients with coupling strength r,
- p is the normalized pump rate,
- A_s is the saturation amplitude,
- dW_1 and dW_2 are independent Gaussian Wiener processes representing vacuum and pump fluctuations for in-phase and quadrature components, respectively.

In Figure 4, we simulate the CIM dynamics for the MAX-CUT problem on the K_{2000} graph. To numerically solve (86) and (87), we apply the Euler-Maruyama integrator Maruyama (1955), yielding the updates

$$c_i(t+1) = c_i(t) + \Big[(-1 + p(t) - c_i(t)^2 - s_i(t)^2)c_i(t) \; + \; r\sum_j J_{ij}\widetilde{c}_j(t)\Big]\Delta_t + \frac{w_i\sqrt{\Delta_t}}{A_s}\sqrt{c_i(t)^2 + s_i(t)^2 + 0.5}, \tag{88}$$

and

$$s_i(t+1) = s_i(t) + \Big[\left(-1 - p(t) - c_i(t)^2 - s_i(t)^2\right) s_i(t)\Big]\Delta_t + \frac{w_i'\sqrt{\Delta_t}}{A_s}\sqrt{c_i(t)^2 + s_i(t)^2 + 0.5}. \tag{89}$$

We have used $\widetilde{c}_i = c_i$ and $\widetilde{c}_i = \mathrm{sign}(c_i)$ (clamped) in the experiments. The pump rate is defined as $p(t) = t/T$, and w_i, w_i' are sampled independently from the standard normal distributions. After runtime T, the i-th spin state is obtained by taking the sign of $c_i(T)$. The hyperparameters r, A_s, Δ_t are tuned for the best results. In Figure 4, we observe that Algorithm 5 for SimCIM (adapted from Goto et al. (2021)) outperforms the simulated version of CIM in Inagaki et al. (2016) based on (86) and (87).

I.4 Spring Ising Algorithm (SIA)

We implemented the Spring Ising Algorithm (SIA) as described in Jiang et al. (2024). The corresponding pseudocode is provided in Algorithm 6. Following Jiang et al. (2024), we set the mass coefficient $m = 1$, the elastic coefficient $k = 0.5$, and the scaling coefficient $\zeta(t)$ to vary linearly from $0.8\zeta_0$ to $10\zeta_0$, where $\zeta_0 = 0.05$ is the base value. The time step Δ is selected within the interval $(0, 1]$ for the best result. The momentum vector $\boldsymbol{p}$ is initialized such that each component p_i is sampled uniformly from the range $(-0.0005, 0.0005)$.

Algorithm 6 Spring Ising Algorithm (SIA)

Input: mass coefficient m, elastic coefficient k, scaling function $\zeta(t)$, step size Δ, total steps N
$n \leftarrow \dim(\mathbf{J})$
$\boldsymbol{q} \leftarrow \mathbf{0}$ ▷ initial displacement
$\boldsymbol{p} \leftarrow \mathrm{Random}(n)$ ▷ small random perturbation in $(-0.0005, 0.0005)$
for $t = 1$ **to** N **do**
  $(\boldsymbol{q}, \boldsymbol{p}) \leftarrow \mathrm{Boundary}(\boldsymbol{q}, \boldsymbol{p})$ ▷ enforce boundary conditions (see Eq. (90))
  $\boldsymbol{q} \leftarrow \boldsymbol{q} + (\Delta/m)\,\boldsymbol{p}$
  $\boldsymbol{p} \leftarrow \boldsymbol{p} - \Delta k\,\boldsymbol{q} + \zeta(t)\,\Delta\,\mathbf{J}\boldsymbol{q}$
end for
Output: $\boldsymbol{s} = \mathrm{sign}(\boldsymbol{q})$

The function $\mathrm{Boundary}(\boldsymbol{q}, \boldsymbol{p})$ enforces the following componentwise constraints on the vectors $\boldsymbol{q}$ and $\boldsymbol{p}$:

$$q_i \leftarrow \begin{cases} \sqrt{2}, & q_i > \sqrt{2}, \\ q_i, & -\sqrt{2} \leqslant q_i \leqslant \sqrt{2}, \\ -\sqrt{2}, & q_i < -\sqrt{2}, \end{cases} \qquad p_i \leftarrow \begin{cases} 2, & p_i > 2, \\ p_i, & -2 \leqslant p_i \leqslant 2, \\ -2, & p_i < -2. \end{cases} \tag{90}$$

I.5 Quantum-inspired Quadratic Annealing (QQA)

We evaluate QQA using the authors' public implementation and its `qqa.anneal` routine Ichikawa & Arai (2025). QQA performs continuous relaxation with an annealed quadratic objective, then rounds the final state. We expose the implementation's annealing and step-size controls to the same adaptive box-search budget used for every solver, and report this tuning time separately from the time of a selected run. No method-specific manual refinement is applied after the common search.

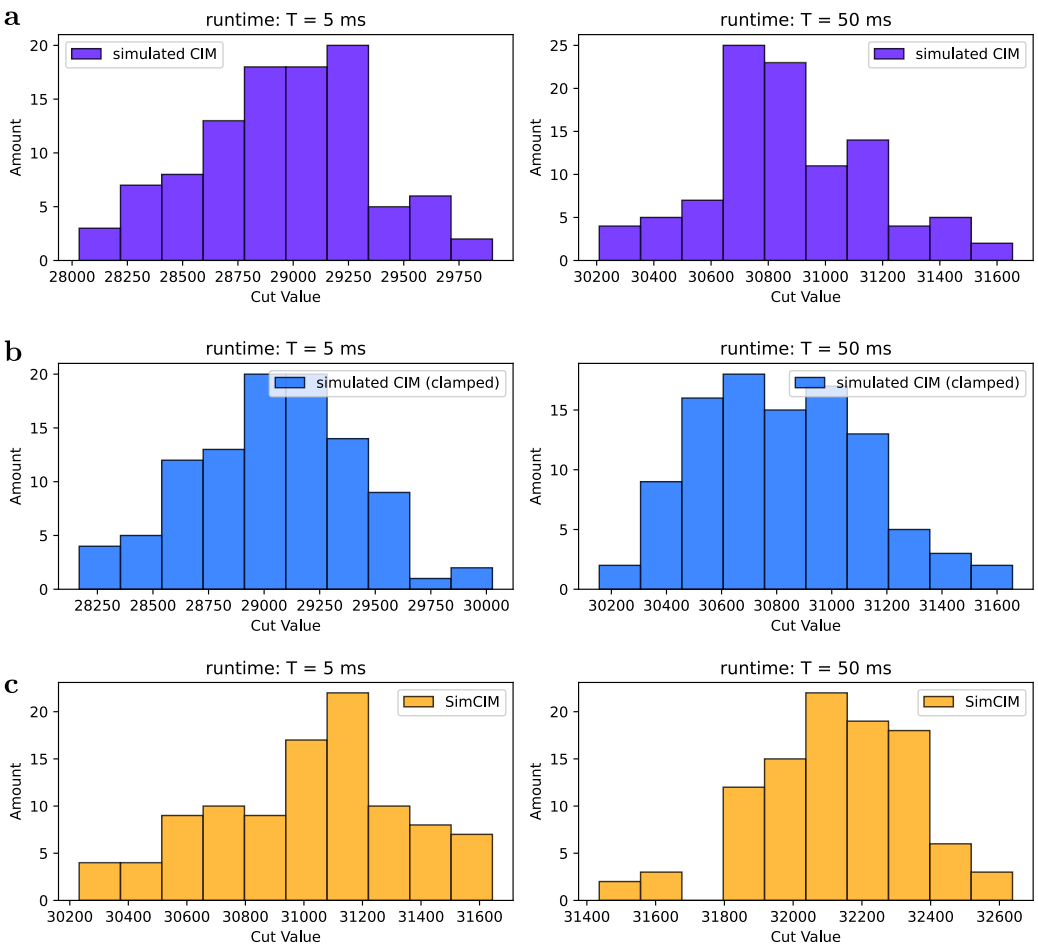


Figure 4: **Comparison of simulated CIM Inagaki et al. (2016) with SimCIM.** Histogram plots of cut values obtained for K_{2000} graph, with 100 different initializations and two runtimes ($T = 5$ ms, 50 ms). **(a), (b)** Cut values obtained by two variants of simulated CIM (from equations (88), (89)). **(c)** Cut values obtained by SimCIM (from Algorithm 5). The results were obtained on a laptop equipped with a 4GB NVIDIA RTX 3050 GPU.

I.6 iSCO

We adapt the reference iSCO implementation distributed with RLSolver Sun et al. (2023). The method draws multiple proposal chains for an Ising objective and returns the best discrete state across the chains. Our implementation evaluates the same zero-field energy (2), retains the reference multi-chain update, and uses the common adaptive box search for temperature and proposal controls. The results in Tables 4 and 5 use the same instances, hardware, run identifiers, and evaluation code as DOCH, ADOCH, and QQA.

J Data Generation

To construct a sparse instance, let $c \in (0, 1]$ be its connectivity fraction, i.e., the expected fraction of active off-diagonal pairs. We generate the strict lower triangle row by row, mirror it to the upper triangle, and keep $J_{ii} = 0$. For row i, we draw the number of active predecessors as $d_i \sim \text{Binomial}(i - 1, c)$ and select d_i distinct columns without replacement. Each active coupling is then sampled uniformly from $\{-511, \ldots, -1, 1, \ldots, 511\}$. This excludes zero and removes the ambiguity in the earlier "signed 9-bit" description. Sampling only the required nonzeros avoids an $\mathcal{O}(n^2)$ sequence of Bernoulli trials.

For the 10^8-spin sparse experiment, the instance is generated once with seed 42. The generator uses deterministic row-chunk offsets of size 10^5, saves the symmetric matrix as a CSR `.npz` file Saad (2003), and all solvers reuse that same file. Thus, solver comparisons do not regenerate or resample the instance. The same row-wise construction is used at smaller sparse scales.

For the ultra-large nominally dense experiment, explicit storage is prohibitively expensive. We therefore use the structured rule $J_{ij} = \sin(ij + \text{seed})$ with seed 100 and generate blocks on demand. This matrix is symmetric and exactly reproducible, but it is a matrix-free structured stress test rather than a generic stored dense random matrix. We accordingly use it only to study implementation scaling and do not interpret the resulting energy as evidence for arbitrary dense Ising instances.

Algorithm 7 Row-wise generation of a sparse coupling matrix in CSR format

Input: size $n \geqslant 1$, connectivity $c \in (0, 1]$, seed $u = 42$, row chunk $B = 10^5$
Initialize: data $\leftarrow [\,]$, column_indices $\leftarrow [\,]$, row_offset array of length $(n + 1)$
row_offset$[1] \leftarrow 0$
for each row chunk $r : r + B - 1$ **do**
  Initialize the PRNG at the deterministic offset associated with (u, r)
  for $i = r$ **to** $\min(r + B - 1, n)$ **do**
    $d_i \sim \text{Binomial}(i - 1, c)$
    $C_i \leftarrow \text{SampleWithoutReplacement}(\{1, \ldots, i - 1\}, d_i)$
    for $j \in \text{sort}(C_i)$ **do**
      column_indices.store(j)
      data.store$\big(\text{Uniform}(\{-511, \ldots, -1, 1, \ldots, 511\})\big)$
    end for
    row_offset$[i + 1] \leftarrow$ row_offset$[i] + d_i$
  end for
end for
$\mathbf{J}_{\text{upper}} \leftarrow \text{CSR}(\text{data, column_indices, row_offset})$
$\mathbf{J} \leftarrow \mathbf{J}_{\text{upper}} + \mathbf{J}_{\text{upper}}^{\top}$
Save $\mathbf{J}$ once in CSR `.npz` format
Output: $\mathbf{J}$

K Large Matrix-Vector Multiplication

For ultra-large matrices, matrix-vector multiplications are expensive and memory-intensive. We therefore partition a procedural dense matrix into $b \times b$ blocks with $b = 10^3$; each block is generated on demand, used for a partial matrix-vector product, and discarded. Sparse CSR products are processed in row chunks of size 10^5. The dense computation is distributed across the stated two to four GPUs. All state vectors, generated blocks, sparse values, and matrix-vector products use FP32. FP64 is used only when evaluating scalar constants in the deterministic matrix-generation rule before casting to FP32. The exact Hamiltonian check in ADOCH reuses the already-computed $\mathbf{J}\boldsymbol{x}$ product rather than performing an additional matrix-vector multiplication.

L Benchmarking Information

The table below lists the Ising models used for additional benchmarking. For each model, we specify the number of spins n, the connectivity fraction c, the coupling distribution, and the GPU hardware. A connectivity of $c\%$ means that an expected fraction $c/100$ of off-diagonal pairs is active.

For large random matrices, we estimate $-\lambda_{\min}(\mathbf{J})$ using the Wigner approximation in (82). The 10^7-spin dense matrix is procedurally generated; the ultra-large instances are synthetic and have no known optimum or independent bound. The longest experiment uses four H100 GPUs for 11 hours, so compute time, energy use, and specialized hardware access are material practical costs. We have matched tuning budgets and use official implementations when available. We therefore report certified small-instance tests, uncertainty statistics, and tuning time separately from the scalability stress tests.

Algorithm 8 Large Matrix-Vector Multiplication

Input: matrix $\mathbf{J} \in \mathbb{R}^{n\times n}$, vector $\boldsymbol{v} \in \mathbb{R}^n$, block size $b \times b$, number of GPUs G
Output: $\boldsymbol{y} = \mathbf{J}\boldsymbol{v}$
Initialize $\boldsymbol{y} \leftarrow \mathbf{0}_n$
// distribute matrix blocks across G GPUs to balance computation
// — begin parallel execution across GPUs —
for $g = 1$ **to** G **do**
  // GPU g processes its assigned blocks
  for each (i, j) block assigned to GPU g **do**
    // dynamically generate block $\mathbf{J}_{i:i+b,\ j:j+b}$
    $\mathbf{J}_{\text{block}} \leftarrow \mathbf{J}[i : i + b,\ j : j + b]$
    $\boldsymbol{v}_{\text{sub}} \leftarrow \boldsymbol{v}[j : j + b]$
    $\boldsymbol{y}[i : i + b] \leftarrow \boldsymbol{y}[i : i + b] + \mathbf{J}_{\text{block}} \cdot \boldsymbol{v}_{\text{sub}}$
    // discard $\mathbf{J}_{\text{block}}$ to free memory
  end for
end for
// — end parallel execution —
Return: $\boldsymbol{y}$

Table 3: Ising model parameters and hardware used for additional benchmarking.

Fig.	n	connectivity	J_{ij}	GPU
7	10^3	fully connected	SK model	NVIDIA jetson nano
8	10^4	fully connected	SK model	NVIDIA jetson nano
9	10^4	1%	$\{-511, \dots, -1, 1, \dots, 511\}$	NVIDIA jetson nano
10	10^5	fully connected	$\{-1, 1\}$	1× RTX 3090, 24 GB
11	10^5	fully connected	SK model	1× RTX 3090, 24 GB
12	10^5	1%	$\{-511, \dots, -1, 1, \dots, 511\}$	1× RTX 3090, 24 GB
13	10^6	fully connected	$\sin(i\,j + \text{seed})$	1× V100, 32 GB
14	10^6	0.1%	$\{-511, \dots, -1, 1, \dots, 511\}$	1× V100, 32 GB
15	10^6	0.01%	$\{-511, \dots, -1, 1, \dots, 511\}$	1× V100, 32 GB
16	10^7	0.001%	$\{-511, \dots, -1, 1, \dots, 511\}$	4× V100, 30 GB
17	$10^4 - 10^5$	variable	variable	1× RTX 3090, 24 GB
18	800	variable	G-set graphs	1× RTX 3050, 4 GB
19	80, 100	variable	Biq Mac graphs	NVIDIA jetson nano
20	2000	fully connected	$\{-1, 1\}$	1× RTX 3050, 4 GB

M Reproducibility and Additional Comparisons

M.1 Common tuning and execution protocol

We use the same adaptive box search as per Algorithm 9, for the hyperparameters of every solver. Starting from the admissible interval given by the corresponding method, we evaluate $m = 3$ uniformly spaced values and refine the interval around the best value. We use $R = 3$ refinement rounds for $n \leqslant 10^4$ and one round for the ultra-large experiments. When a method has multiple hyperparameters, this operation is applied coordinate-wise using the same number of evaluations per control. The reported tuning time includes all evaluations made by this search, whereas runtime is the time of the selected solver run. The final discrete energy is included in runtime; intermediate diagnostic plots exclude only the optional logging of energy at every iteration.

Unless stated otherwise, the random seed is 42, computation is FP32, and 100 initializations are used for stochastic summaries. The QQA/iSCO comparison uses 100 run seeds in the range 42–143. The ultra-large 10^7-spin dense experiment alone uses seed 100.

M.2 QQA and iSCO

For a spin vector $\boldsymbol{s}$, we report the 1-flip score

$$\text{score}_{1\text{f}}(\boldsymbol{s}) = \boldsymbol{s}^\top \text{sign}(\mathbf{J}\boldsymbol{s}) \leqslant n. \tag{91}$$

Algorithm 9 Adaptive box search used for all solvers

Input: interval $[t_{\min}, t_{\max}]$, resolution $m = 3$, rounds R
$\mathcal{V} \leftarrow \text{linspace}(t_{\min}, t_{\max}, m)$
for $r = 1$ **to** R **do**
  $t^\star \leftarrow \arg\min_{t \in \mathcal{V}} \text{Energy}(\text{Solver}(t))$
  if $t^\star$ is the left endpoint **then**
    $\mathcal{V} \leftarrow \text{linspace}(\max\{t^\star/2, t_{\min}\}, t^\star, m)$
  else if $t^\star$ is the right endpoint **then**
    $\mathcal{V} \leftarrow \text{linspace}(t^\star, \min\{2t^\star, t_{\max}\}, m)$
  else
    $\mathcal{V} \leftarrow \text{linspace}(\text{left}(t^\star), \text{right}(t^\star), m)$
  end if
end for
Output: $t^\star$

A larger value means that more spins align with their local fields, and the maximum n is attained at a 1-flip local optimum (i.e. flipping any one spin doesn't decrease the Ising energy). Each objective ratio is normalized by the best corresponding objective among QQA, iSCO, DOCH, and ADOCH, so that 1 is best. These are the relative comparison ratios, not approximation certificates or ratios to the unknown ground-state energy. All entries are based on 100 independent runs on a 4GB NVIDIA RTX 3050 GPU. QQA uses its official library Ichikawa & Arai (2025); the corrected 10^4-spin iSCO entries use the published multi-chain configuration with sufficiently long chains Sun et al. (2023).

Table 4: QQA and iSCO comparison on 1000-spin instances. Tuning and selected-run times are reported separately.

Instance	Solver	Mean ratio	Best ratio	Mean 1-flip	Best 1-flip	Tune (s)	Runtime (s)
K_{1000} MaxCut	QQA	0.959	0.943	972.3	**986**	2.644	0.106
	iSCO	0.392	0.483	380.2	514	0.935	**0.104**
	DOCH	**1.000**	**1.000**	956.9	980	0.183	0.181
	ADOCH	0.994	0.988	**974.8**	982	**0.115**	0.106
SK_{1000} Ising	QQA	0.951	0.947	971.7	980	2.640	0.106
	iSCO	0.637	0.683	699.0	798	0.929	**0.104**
	DOCH	**1.000**	**1.000**	974.5	984	0.181	0.181
	ADOCH	0.984	0.992	**974.9**	**986**	**0.115**	0.106

Table 5: QQA and corrected multi-chain iSCO comparison on 10 000-spin instances.

Instance	Solver	Mean ratio	Best ratio	Mean 1-flip	Best 1-flip	Tune (s)	Runtime (s)
$K_{10\,000}$ MaxCut	QQA	0.924	0.920	9633.4	9650	59.70	6.48
	iSCO	0.892	0.906	9312.7	9450	9.29	**1.13**
	DOCH	**1.000**	**1.000**	**9710.4**	**9712**	2.04	1.99
	ADOCH	0.994	0.993	9682.6	9686	**1.40**	1.17
$SK_{10\,000}$ Ising	QQA	0.924	0.920	9600.8	9644	60.45	6.66
	iSCO	0.884	0.898	9241.8	9380	9.35	**1.12**
	DOCH	**1.000**	**1.000**	**9720.7**	**9730**	2.04	2.00
	ADOCH	0.993	0.992	9672.9	9686	**1.40**	1.16

M.3 Optimized SA and certified optima

Table 6 compares the optimized SA implementation of Isakov et al. Isakov et al. (2015). SA is allowed to consume as many full sweeps as fit in the same approximately 0.5–2 s wall-clock budget as DOCH.

We further measure the average single-run time to reach a certified global optimum on small instances. The optima were obtained by exhaustive enumeration, so unlike the large-instance ratios, the target in Table 7 is independent of the compared solvers.

Table 6: Equal-wall-clock comparison with optimized SA. Ratios are relative to the best objective found in the comparison, not to a certified optimum.

Instance	DOCH	Optimized SA Isakov et al. (2015)
K_{1000} Ising	1.000	0.966
K_{1000} MaxCut	1.000	0.984
SK_{1000} Ising	1.000	0.987
SK_{1000} MaxCut	1.000	0.982
K_{2000} MaxCut	1.000	0.985

Table 7: Average single-run time to a certified optimum (ms).

Instance	SA	DOCH	ADOCH
Rudy pm1 ($n = 24$)	1.12	0.05	0.07
Rudy w05 ($n = 24$)	4.03	0.10	0.14
K_{22}	0.52	0.04	0.07
SK ($n = 24$)	3.15	0.07	0.08

M.4 Planted signed-bipartite instances

Let $B = \boldsymbol{p}\boldsymbol{q}^\top$, with $p_i, q_i \in \{-1, 1\}$, and construct

$$\mathbf{J} = \begin{pmatrix} \mathbf{0} & B \\ B^\top & \mathbf{0} \end{pmatrix}.$$

For balanced parts, the planted spin $(\boldsymbol{p}, \boldsymbol{q})$ has the certified ground-state energy $\mathcal{E}^\star$ (see Table 8). Each solver is tuned for approximately 0.05 s per trial using $m = R = 3$, then run for approximately 0.1 s from each of 100 initializations with seed 42.

Table 8: Ground-state recovery on planted rank-one signed-bipartite instances. "Best ratio" is $(\min \mathcal{E})/\mathcal{E}^\star$.

n	Solver	Tuned η	$\mathcal{E}^\star$	Smallest $\mathcal{E}$	Mean $\mathcal{E}$	Best ratio	Mean runtime (s)
10	DOCH	1.003	-25	-25	-20.84	1.000	0.146
10	ADOCH	1.005	-25	-25	-20.84	1.000	0.103
20	DOCH	1.001	-100	-100	-69.84	1.000	0.143
20	ADOCH	0.010	-100	-100	-14.00	1.000	0.103
50	DOCH	0.500	-625	-625	-625.00	1.000	0.146
50	ADOCH	1.005	-625	-625	-318.08	1.000	0.103
100	DOCH	0.500	-2500	-2500	-2500.00	1.000	0.148
100	ADOCH	1.005	-2500	-2500	-680.08	1.000	0.103
1000	DOCH	0.500	-250000	-250000	-247499.84	1.000	0.186
1000	ADOCH	0.010	-250000	-250000	-5000.00	1.000	0.106

N Additional Benchmarking Figures

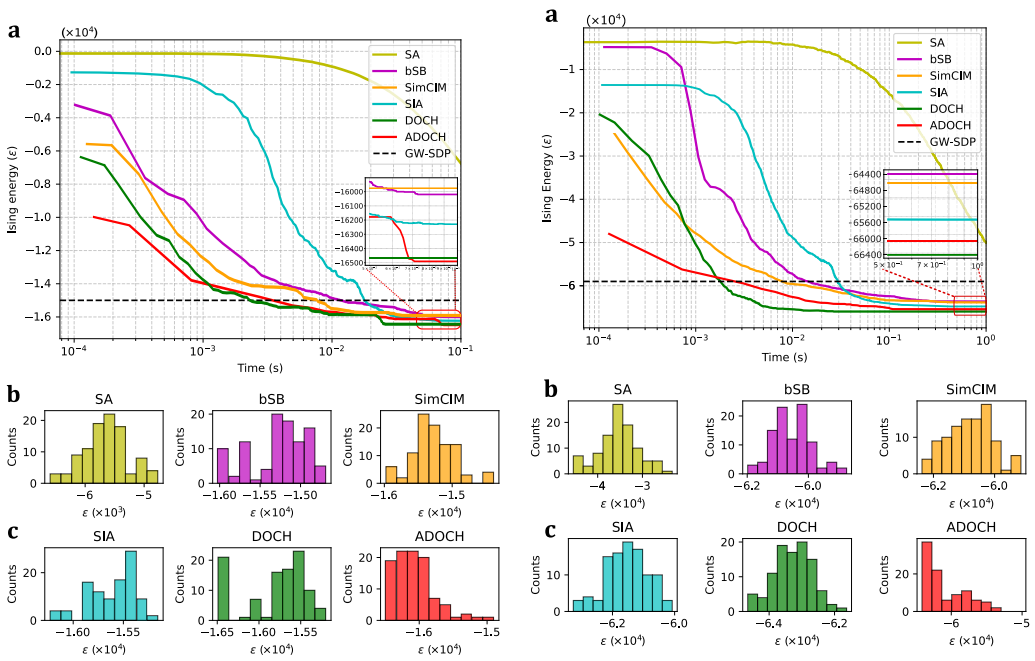


Figure 5: **Ising energy vs. runtime for the 10^3-spin SK model (left) and K_{2000} graph (right).** **(a)** DOCH and ADOCH compared with the original reference SA trace, bSB Goto et al. (2021), SimCIM Goto et al. (2021), and SIA Jiang et al. (2024). The final equal-wall-clock comparison uses optimized SA Isakov et al. (2015) and is reported in Table 6. Solid curves show the best (left) and mean (right) energy over 100 runs. Dashed black lines indicate the best GW-SDP energy. **(b), (c)** Energy histograms over 100 runs with different initializations. Experiments are run on an NVIDIA RTX 3050 laptop.

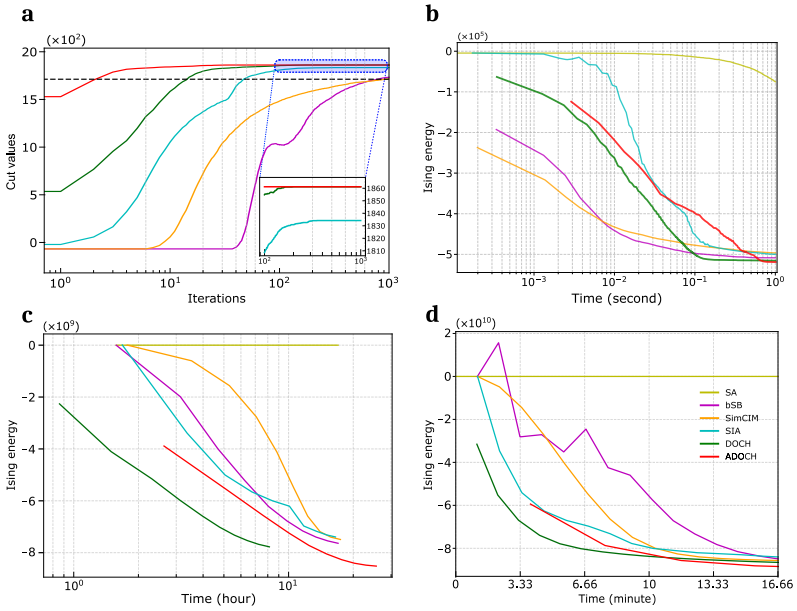


Figure 6: **Solver performance across scales.** **(a)** MAX-CUT on G_{10} (800 nodes): mean cut vs. iterations over 100 runs; Jetson Nano. The dashed line is GW-SDP. **(b)** 10^4-spin SK model: energy vs. time (log scale); Jetson Nano. **(c)** Fully connected 10^7-spin model ($J_{ij} = \sin(ij + \text{seed})$): energy vs. time; four H100 GPUs. **(d)** Sparse 10^8-spin model (0.00001% connectivity, $J_{ij} \in \{-511, \ldots, -1, 1, \ldots, 511\}$): energy vs. time (minutes); two H100 GPUs. Panels (c)–(d) are empirical scalability stress tests without certified optima.

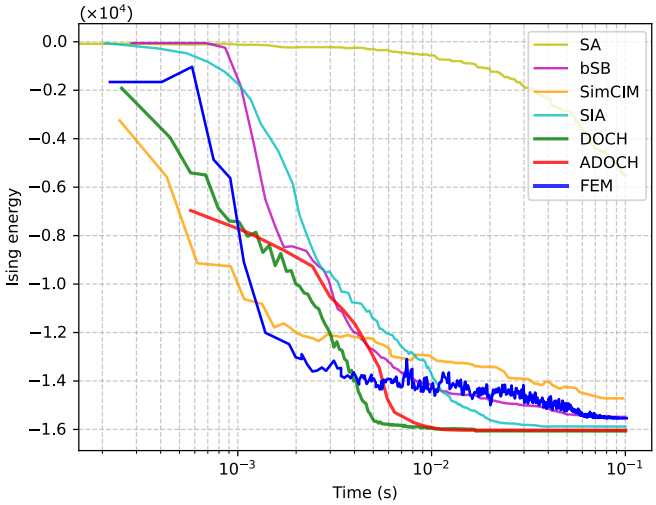


Figure 7: Benchmarking results for a 10^3 spin Sherrington–Kirkpatrick (SK) model. The plot shows the observed Ising energy as a function of computation time (log scale) for the compared solvers. DOCH and ADOCH reach lower energies earlier than the baselines in this run. The Free Energy Machine (FEM) was evaluated using its official implementation Shen et al. (2025) on an NVIDIA Jetson Nano.

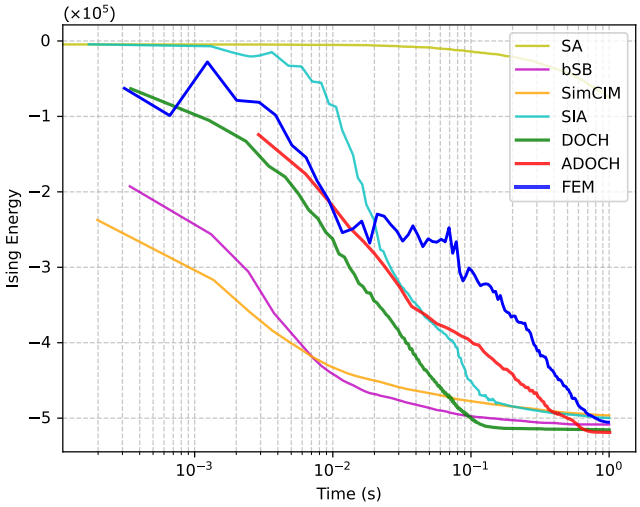


Figure 8: Benchmark results for a 10^4 spin SK model. The plot shows Ising energy versus computation time on a logarithmic scale for the compared solvers. DOCH and ADOCH obtain the best observed time–energy trade-off in this experiment. FEM results are obtained using the official implementation Shen et al. (2025), executed on an NVIDIA Jetson Nano.

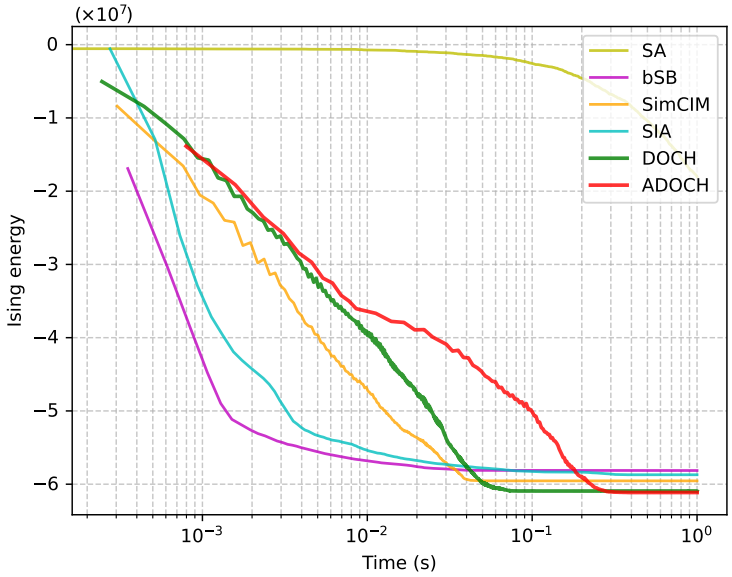


Figure 9: Benchmark results for a large-scale sparse Ising instance with 10^4 spins and 1% connectivity. Nonzero couplings are drawn uniformly from $J_{ij} \in \{-511, \ldots, -1, 1, \ldots, 511\}$. Ising energy is plotted against computation time on a logarithmic scale. The results compare the observed time–energy trajectories under low connectivity and discrete weights. All experiments were executed on an NVIDIA Jetson Nano.

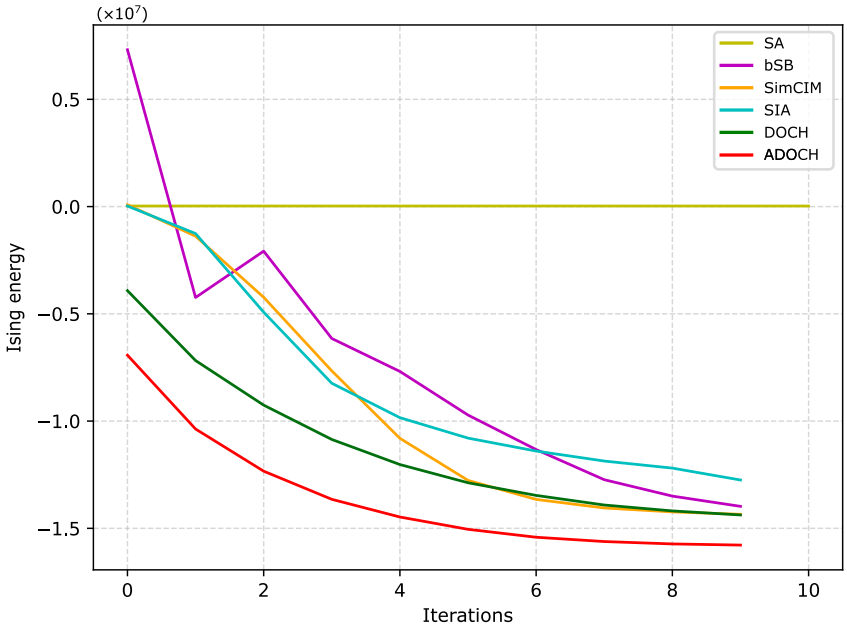


Figure 10: Performance comparison on K100000 (a fully connected Ising model with 10^5 spins and binary couplings $J_{ij} \in \{-1, +1\}$ drawn with equal probability). **Top:** Ising energy evolution over time (log scale). **Bottom:** Ising energy as a function of iteration count. These results demonstrate the rapid energy descent of our solvers on large and dense instances. All experiments were performed on a single NVIDIA RTX 3090 GPU.

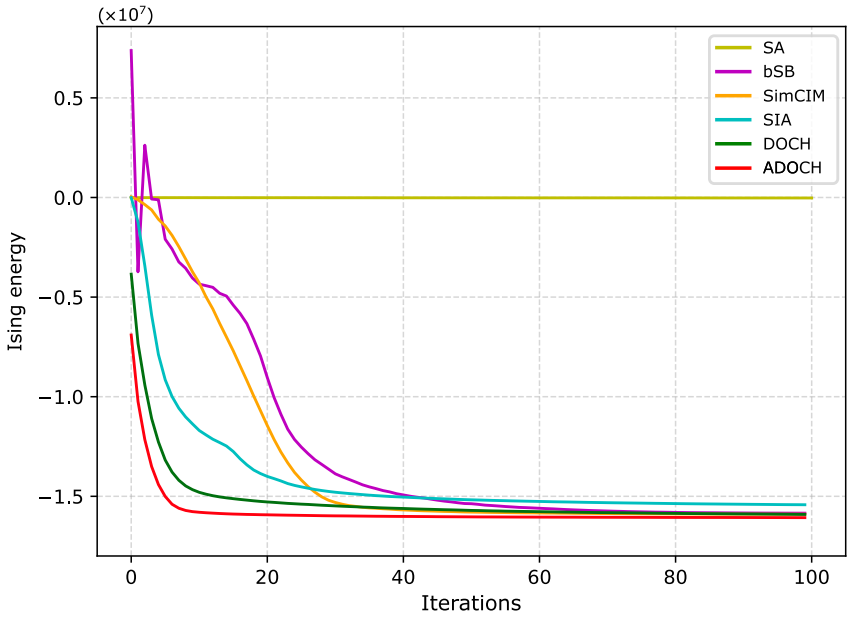


Figure 11: Performance comparison on a fully connected 10^5 spin SK. The Ising energy is plotted against the iteration count, up to 100 iterations. We used a single NVIDIA RTX 3090 GPU for the computations.

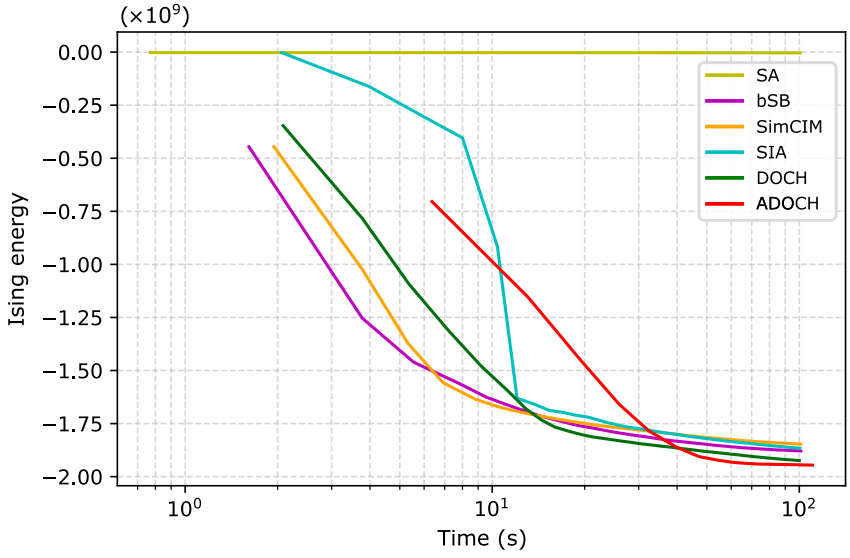


Figure 12: Performance comparison of Ising energy for a 10^5 spin Ising model with 1% connectivity. Nonzero couplings are drawn uniformly from $J_{ij} \in \{-511, \ldots, -1, 1, \ldots, 511\}$. The results are for a single NVIDIA RTX 3090 GPU.

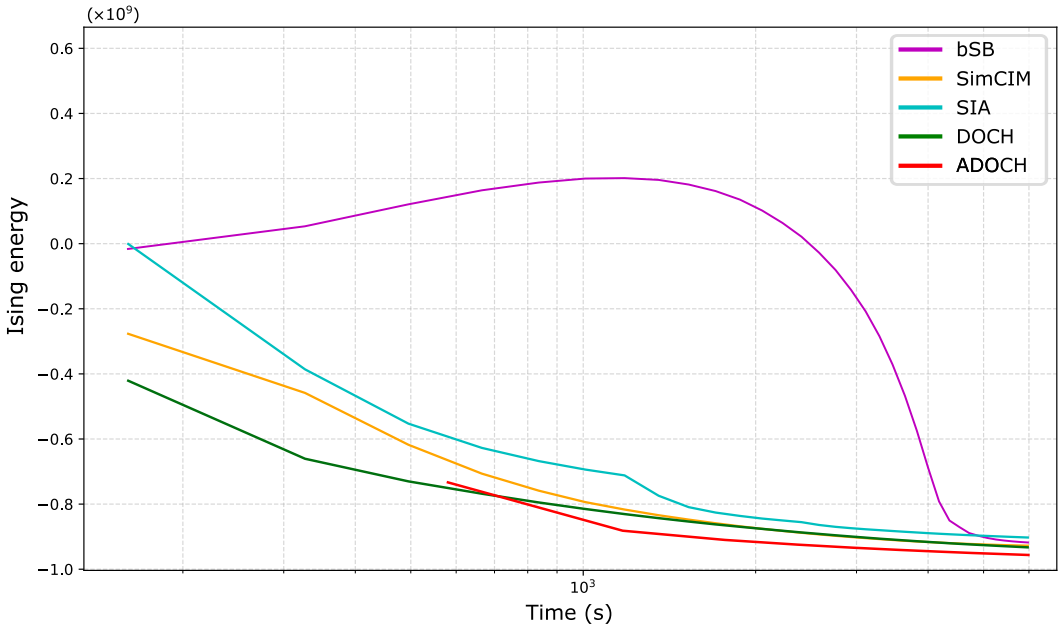


Figure 13: Time evolution of Ising energy for a 10^6 spin fully connected Ising model with structured couplings $J_{ij} = \sin(i\,j + \text{seed})$. The results are for a single NVIDIA V100 GPU.

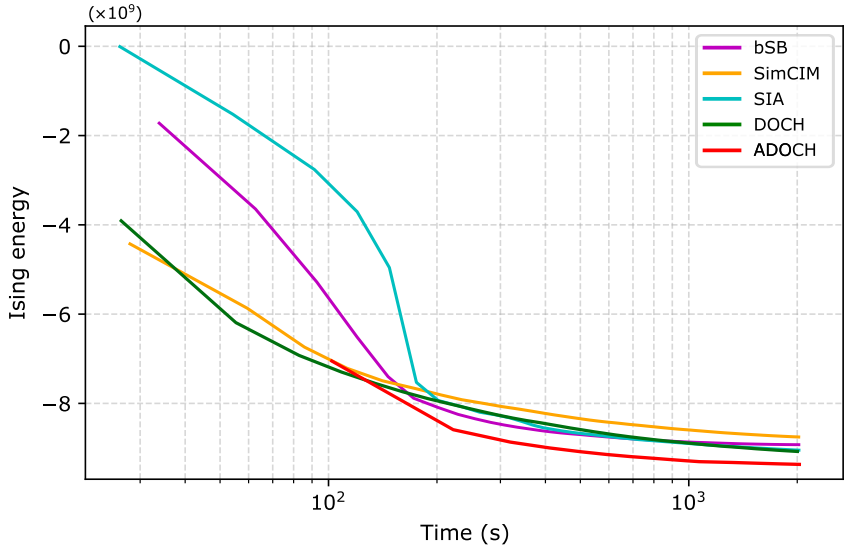


Figure 14: Time evolution of Ising energy for a 10^6 spin sparsely connected Ising model, where 0.1% of the couplings are nonzero. Each active coupling is sampled uniformly from $J_{ij} \in \{-511, \ldots, -1, 1, \ldots, 511\}$. All solvers were run on a single NVIDIA V100 GPU.

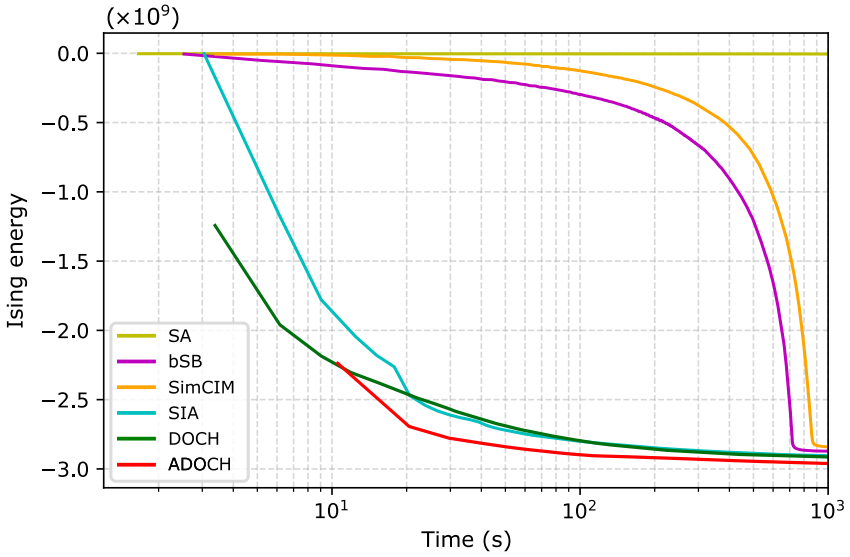


Figure 15: Time evolution of Ising energy for a 10^6 spin extremely sparse Ising model with 0.01% connectivity. Each nonzero coupling is drawn uniformly from $J_{ij} \in \{-511, \ldots, -1, 1, \ldots, 511\}$. All solver experiments were executed on a single NVIDIA V100 GPU.

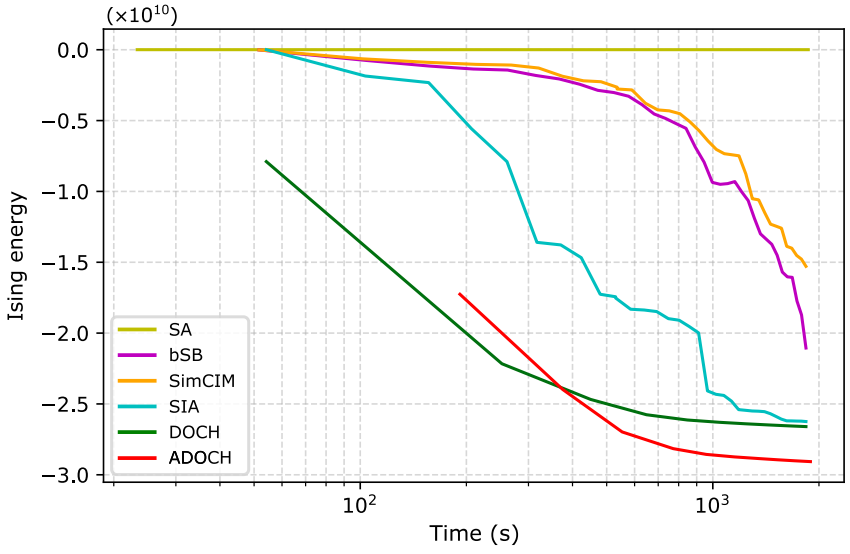


Figure 16: Benchmarking performance on a 10^7 spin sparse Ising model with 0.001% connectivity. Nonzero couplings are drawn uniformly from $J_{ij} \in \{-511, \ldots, -1, 1, \ldots, 511\}$. The vertical axis represents the Ising energy, while the horizontal axis (on a log scale) denotes computation time in seconds. Each solver was executed using four NVIDIA V100 GPUs.

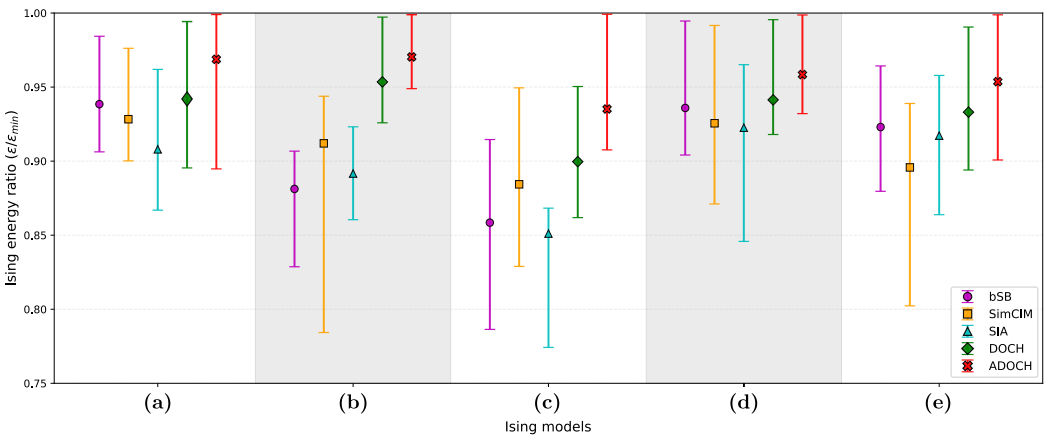


Figure 17: **Error-bar analysis for medium and large-scale Ising models.** The plots show the Ising energy ratio (energy obtained by a solver normalized by the best energy across solvers) after a fixed runtime T. These are within-comparison ratios rather than approximation certificates. For each solver, the mean energy ratio is shown, with error bars indicating the maximum and minimum energy values over 100 random initializations. (a) 10^4-spin SK model, runtime $T = 1$ s; (b) 10^4-spin Ising model with 1% connectivity and nonzero $J_{ij} \in \{-511, \ldots, -1, 1, \ldots, 511\}$, runtime $T = 1$ s; (c) $K100000$ graph, $T = 10$ s; (d) 10^5-spin SK model, $T = 10$ s; (e) 10^5-spin Ising model with 1% connectivity and nonzero $J_{ij} \in \{-511, \ldots, -1, 1, \ldots, 511\}$, runtime $T = 10$ s. All computations were performed on a single NVIDIA RTX 3090 GPU.

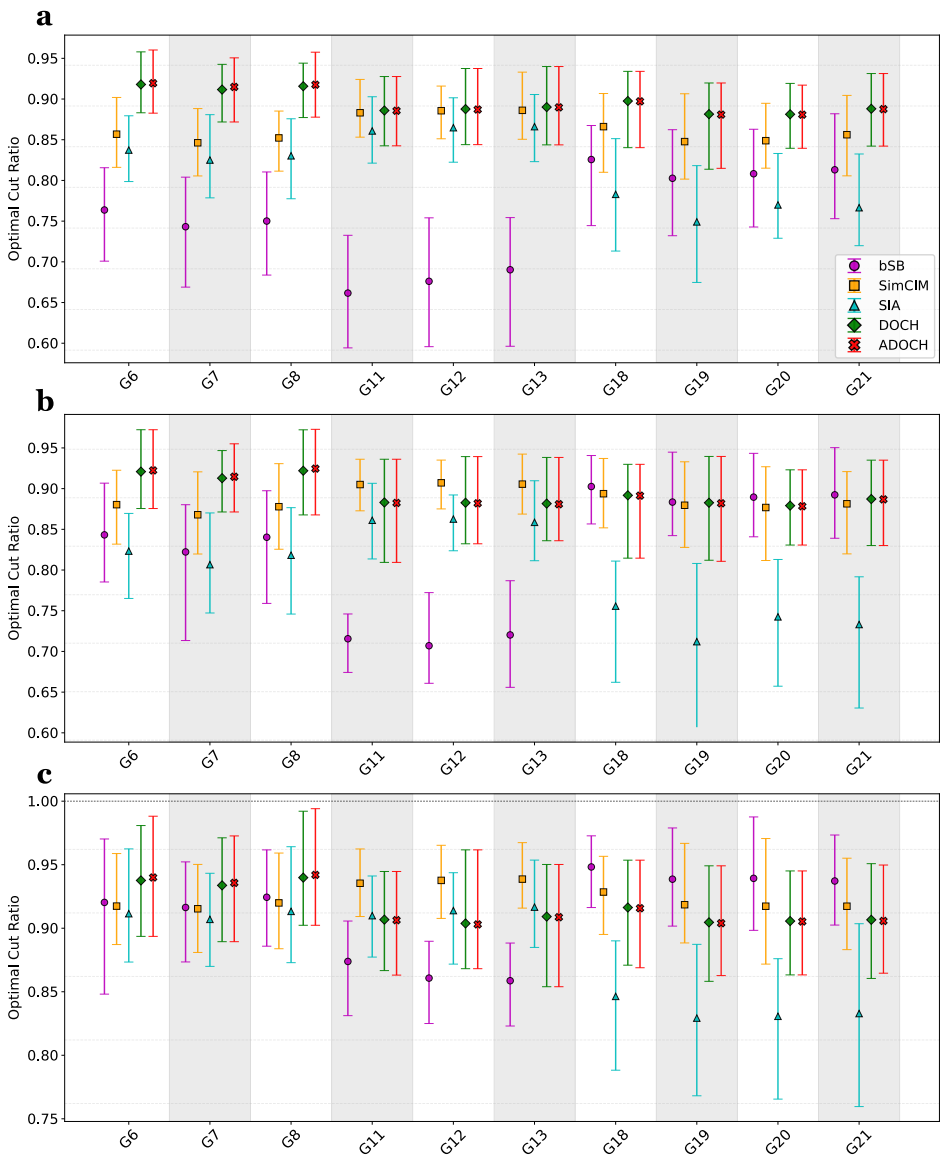


Figure 18: **MAX-CUT on G-set graphs.** The mean cut ratio (mean cut normalized by the cut value reported in Ma & Hao (2017)) is shown for G-set graphs (random: G_6, G_7, G_8; toroidal: G_{11}, G_{12}, G_{13}; planar: $G_{18}, G_{19}, G_{20}, G_{21}$). Panels **(a)**, **(b)**, and **(c)** show the results after 5, 10, and 100 iterations. The error bars indicate the maximum and minimum values across 100 random initializations, respectively. The computations were performed on an NVIDIA RTX 3050 GPU laptop.

a

graph	node	range of J_{ij}	sparsity	solvers	avg TTS (ms)
w05_100.6	100	{-10, …, +10}	52.42%	SA	18.36 ± 0.90
				bSB	19.64 ± 0.39
				SimCIM	17.67 ± 0.42
				SIA	22.33 ± 0.58
				DOCH	1.01 ± 0.09
				ADOCH	0.85 ± 0.38
pm1d_80.4	80	{-1, 0, +1}	1.01%	SA	17.80 ± 0.28
				bSB	19.43 ± 0.36
				SimCIM	17.32 ± 0.24
				SIA	21.70 ± 0.32
				DOCH	0.86 ± 0.09
				ADOCH	0.67 ± 0.29
w05_100.9	100	{-10, …, +10}	52.42%	SA	18.09 ± 0.59
				bSB	19.61 ± 0.28
				SimCIM	17.61 ± 0.18
				SIA	22.14 ± 0.39
				DOCH	1.00 ± 0.09
				ADOCH	1.33 ± 0.61
w05_100.2	100	{-10, …, +10}	52.48%	SA	17.75 ± 0.25
				bSB	19.65 ± 0.12
				SimCIM	17.59 ± 0.11
				SIA	22.29 ± 0.38
				DOCH	0.98 ± 0.05
				ADOCH	0.83 ± 0.27
w09_100.7	100	{-10, …, +10}	52.22%	SA	18.13 ± 0.19
				bSB	19.80 ± 0.41
				SimCIM	17.62 ± 0.16
				SIA	22.29 ± 0.83
				DOCH	0.98 ± 0.03
				ADOCH	1.18 ± 0.44

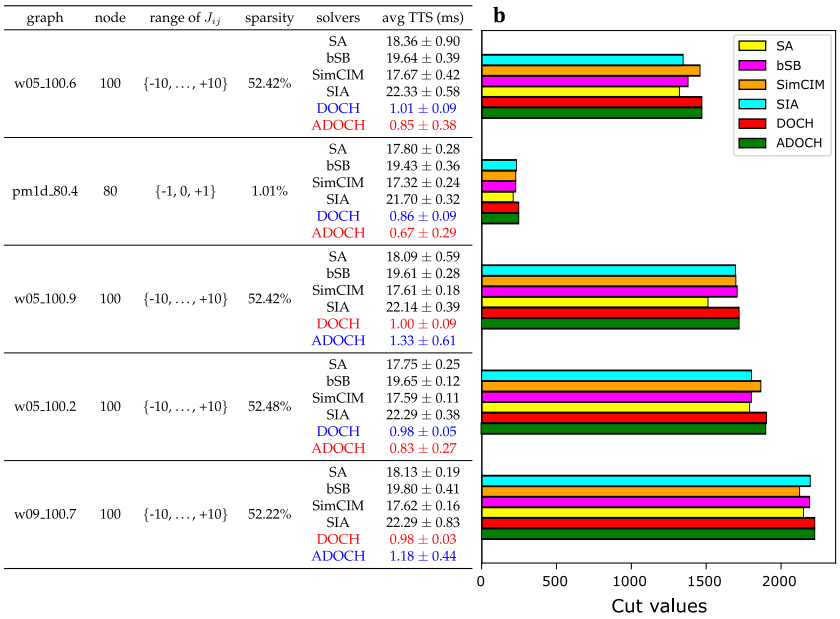


Figure 19: **MAX-CUT performance on Biq Mac benchmark graphs.** Random integer-weighted instances are taken from the Biq Mac Library Wiegele (2007). **(a)** Graph statistics together with the average time-to-solution (Avg TTS) required by each solver to reach 99% of the best cut value obtained by 10^5 iterations of the reference simulated annealing (SA); the best and second-best Avg TTS on each instance are highlighted in red and blue, respectively. This SA-defined threshold is a diagnostic target, not a certified optimum or an independently defined time-to-target. **(b)** Best cut values attained after 1000 solver iterations across all compared methods. DOCH and ADOCH reach the diagnostic target fastest on the displayed instances and obtain best or near-best cuts among the methods shown. All experiments are conducted on an NVIDIA Jetson Nano platform.

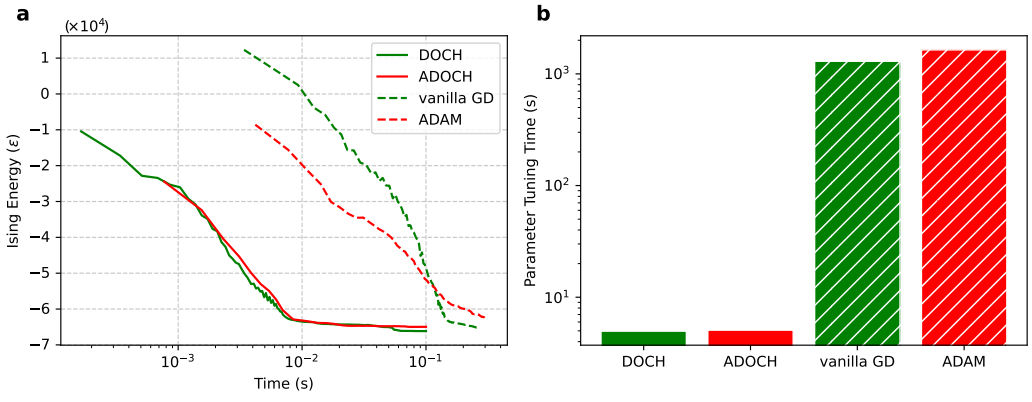


Figure 20: **Comparison of DCP-based Ising solvers (DOCH, ADOCH) with gradient-based methods (vanilla GD, Adam) on the K_{2000} graph.** **(a)** Observed Ising energy versus time after parameter tuning and excluding optional per-iteration energy logging. **(b)** Total parameter-search time for DOCH, ADOCH, vanilla GD, and Adam. The adaptive search is described in Algorithm 9; gradient step sizes and Adam Kingma & Ba (2015) momentum parameters are searched in $[0.001, 1)$. The panel reports the resulting tuning costs under this common procedure.