Neural Deep Operator Networks Representation of Coherent Ising Machine Dynamics

Arsalan Taassob[*, †] **Aaron Lott**[*] **Davide Venturelli** [*]

Abstract

Coherent Ising Machines (CIMs) are optical devices that employ parametric oscillators to tackle binary optimization problems, whose simplified dynamics are described by a series of coupled ordinary differential equations (ODEs). In this study, we setup a proof-of-concept experiment to learn the deterministic dynamics of CIMs via the use of neural Deep Operator Networks (DeepONet). After training successfully the systems over multiple initial conditions and problem instances, we benchmark the comparative performance of the neural network versus the simulated ODEs on solving fully-connected quadratic binary optimization problems. In our tests, the network is capable of delivering solutions to the optimization problems of comparative quality to the exact dynamics up to 175 spins. The CIM model used is very simple with respect to the state-of-art, but we do not identify roadblocks to go further: given sufficient training resources more sophisticated CIM solvers could successfully be represented by a neural network at a large scale.

1 Introduction

As reviewed in [1], there is great interest in hardware implementation of physics dynamics capable of converging to the low-energy configuration of disordered Ising spin models. More specifically, many combinatorial problems of industrial relevance can be conveniently mapped to finding the ground-state of an instance k of the following Hamiltonian:

$$H_k = \sum_{ij} J_{ij}^{(k)} s_i s_j \tag{1}$$

where J_{ij} are real coefficients and s_i are binary ± 1 variables[3]. One of such *Ising machines* employs the physics of a train of laser pulses coupled with Optical Parametric Oscillators in a loop configuration where carefully designed delay lines (or optoelectronic measurement-feedback mechanisms) can allow for controllable interference of the traveling signals. It's the so-called (optically) *Coherent Ising Machine* (CIM) which has been the subject of a very large number of embodiments and variations [3]. Some of these generalizations are arguably intractable to model, due to quantum effects. However, from a mathematical perspective, once all simplifications are performed, the behavior of one of the simplest noiseless CIM can be represented by interconnected sets of regular differential equations (ODEs):

$$\frac{dx_i(t)}{dt} = (p-1)x_i(t) - x_i(t)^3 + \gamma t \sum_i J_{ij} x_i(t) \tag{2}$$

where x_i is the (quadrature) amplitude of the ith OPO mode (spin), p is the photon pump rate, and γ is a speed factor that determines the rate of growth of the feedback contribution that couples the

[*]USRA Research Institute for Advanced Computer Science, Mountain View, CA, USA

[†]North Carolina State University, Raleigh, NC, USA

[3]Usually the Ising model includes local terms such as $\sum_i h_i s_i$, which are crucial for applied mappings. However, the ground state of these models can always be mapped to the ground state of a quadratic form such as (1) by the introduction of one auxiliary spin [2].

37th First Workshop on Machine Learning with New Compute Paradigms at NeurIPS 2023(MLNPCP 2023).

spins, which linearly increases up to a final value ε reached at time $t = T$. The spin variables to be used for the evaluation of the Ising model cost function are recovered by the binary projection $s_i(t) = sign[x_i(t)]$[4] artificially performed in post-processing. The use of the CIM model to solve Ising problems is motivated by an associated Lyapunov dynamics where each variable is driven adiabatically through a bifurcation after which it becomes effectively a binary spin [4][5].

Motivation for using Neural networks for Ising optimization and Neural Operators

The idea of using neural networks to solve combinatorial optimization problems dates back to Hopfield and Tank [6], but it is benefiting from a recent revival from the field of reinforcement learning [7] as well as from physics-inspired graph neural networks [8, 9]. One intriguing approach, which has not appeared in the literature so far, is to see if a neural network could learn the *operator* consisting of the differential equations of a system that solves a combinatorial optimization problem, such as the CIM. Neural Operators [10] are a rather novel advance allowing for a neural network to be trained to approximate a functional and have shown great promise to help solve inverse problems and to replace multi-physics simulations of PDEs with less time and energy-intensive data-driven approach [11]. This workshop paper investigates the proof-of-concept question of whether we can use these neural operator methods to simulate Eq.(2). In addition to the simulation of the actual continuous dynamics, we try to judge whether the resulting trained network could be a competitive Ising solver, meaning if the postprocessed bitstrings after the neural inference are of quality even when the internal dynamics is not faithfully reproduced.

In terms of neural architecture, we choose to focus on the paradigm of *Deep Operator Networks* (DeepONet) [12] which appears scalable and easily deployable to simulate coupled ODEs.

Problem setup, metric of success, and CIM parameter tuning

We select to benchmark spin-glass Ising problems of the Sherrington-Kirckpatrick type (i.e. in Eq.(1), J_{ij} is taken to be randomly drawn from a Gaussian distribution), which are a well studied family of instances [13]. We conducted a preliminary scan of which values of p and γ minimize on average the cost function H as a function of n, discovering that a good set of fixed parameters for $n = 20$-200 is $p = 0.7$ and $\gamma = 10$ (we use $p = 0.4, 0.5, 0.6, 0.65$ respectively for $n = 3, 5, 10, 15$).

1.1 DeepONets for CIM ODE Simulation

Since p and γ are fixed, the output of the CIM equation is exclusively determined by two inputs: the $n(n-1)/2$ problem specifications coefficients $\{J_{ij}\}$ and the n initial conditions: $\{x_i(0)\}$. In Figure 1, we consider a DeepONet architecture consisting of three networks: one network accepting initial conditions as input, one network accepting the time variable t, and one network processing the J_{ij} coefficients. The outputs of the three networks are combined linearly to produce a single output vector, $\tilde{x}$ which is trained against the loss function taken to be the MSE with respect to the exact $x_i(t + \delta t)$, computed via the exact equations Eq.(2). The network is then trained via back-propagation on N_J problem instances, using R_0 initial conditions, to evolve for $T/\delta t$ points (for our experiments $\delta t = 0.005$ for a total of 400 time points). We use $N_J = 10$ and $R_0 = 20$ for training (for a total of 80,000 data samples) and $N_J = 20$ and $R_0 = 50$ for testing. All simulations presented in this study were conducted in the Google Colab cloud-based environment (Python version 3.10.12 and TensorFlow 2.13.0). Adam optimizer was chosen to strike a balance between training convergence speed and stability. A batch size of 1000 samples per iteration was selected randomly, allowing for efficient data processing and GPU resource utilization. Moreover, a learning rate of 0.001 was adopted to finely adjust model weight updates during training, which proceeds through 1500 epochs. These specific training parameters were meticulously determined through iterative experimentation, ensuring an effective and efficient neural network training process.

In Figure 2 we show the dynamical trajectories of six $x_i(t)$ variables (randomly selected among the n) for three cases selecting a random problem instance and a random initial condition for $n = 20, 40, 100$, for illustrative purposes. In dashed we report the DeepONet outputs: it is clear that as the system size increases, the network increasingly struggles to reproduce the exact solutions of the ODEs. However, while the simulation of the entire operator would target the reconstruction of the $x(t)$ at an arbitrary time, for the purpose of combinatorial optimization, a trained DeepOnet task

[4]Note that there is an implicit dependence of these functions by the set of J_{ij} that specify the problem instance k and by the initial conditions which will be specified by the index r.

[5]It should be noted that it has been shown that the CIM dynamics is also useful to solve non-binary, quadratic, non-convex problems [5].

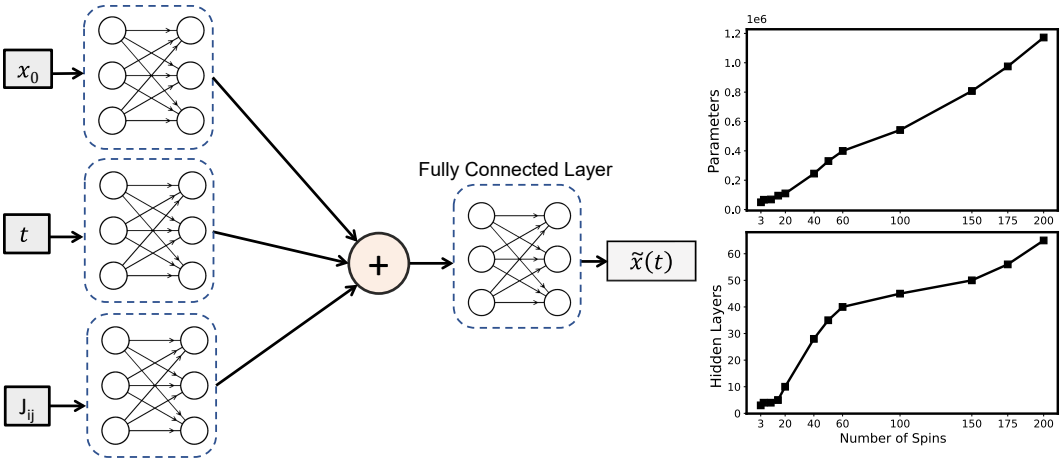

Figure 1: DeepONet architecture used for this study. The panels on the right illustrate the scaling of the complexity of the DeepONet as illustrated by the millions of weights to be trained as well as the hyperparameters for the neural networks accepting x_0 and J_{ij} which need to be adjusted with problem size. Numbers are determined by our empirical fine-tuning. For illustration, we showcase the architectural hyperparameters of the networks for $n = 100$ in Table 1.

Table 1: Parameters for $n = 100$

Layers	1	2	3	4	5	6	7	8	9	...	12	13	...	21
Branch net	64	32	32	32	32	32	32	32	N/A	...	N/A	N/A	...	N/A
Trunk net 1	64	32	32	N/A	N/A	N/A	N/A	N/A	N/A	...	N/A	N/A	...	N/A
Trunk net 2	64	32	32	32	32	32	32	32	32	...	32	N/A	...	N/A
Fully connected dense layer	128	128	128	128	128	128	128	128	128	...	128	128	...	128

would be to deliver the binary spins s_i. For the three cases presented, the final discrete solution is correctly represented.

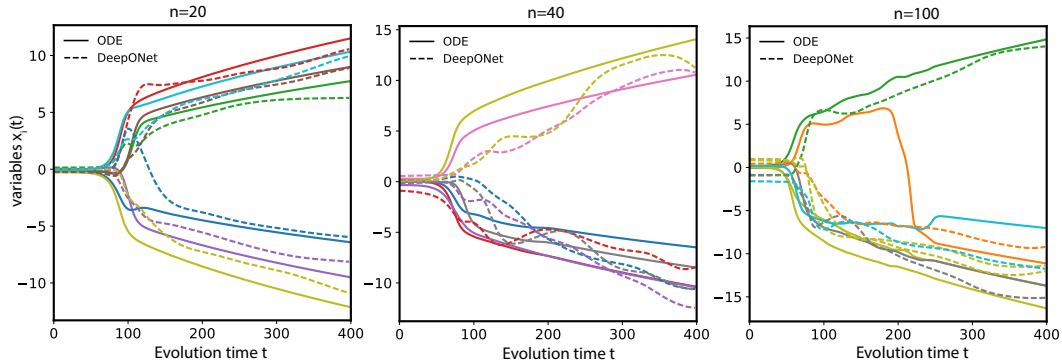

Figure 2: Example spin variable trajectories for three Ising problems at different sizes, obtained by integrating the ODEs in Eq.(2). The bifurcation is complete when the sign of the x_i variables stops changing. Dashed lines indicate the corresponding $\tilde{x}_i$ variables simulated by a DeepONet trained on separate instances of the same size.

In order to differentiate between our two objectives of evaluating the simulation fidelity and the optimization performance, we consider two metrics of success, which we will evaluate as a function of problem size n: one corresponding to the instantaneous mean square error per spin *MSE* and the

other to the average *approximation ratio AR* of the optimization problem[6], i.e.

$$\langle MSE \rangle (n,t) = \frac{1}{nN_J R_0} \left[\sum_k^{N_J} \sum_r^{R_0} \sum_i^n [x_i^{(k,r)}(t) - \tilde{x}_i^{(k,r)}(t)]^2 \right], \tag{3}$$

$$\langle AR \rangle (n) = \frac{1}{N_J R_0} \sum_k^{N_J} \left(\frac{\sum_r^{R_0} \min_t \left[\sum_{ij} J_{ij} sign[\tilde{x}_i^{(k,r)}(t)] sign[\tilde{x}_i^{(k,r)}(t)] \right]}{\min_{s_i,s_j} \left[\sum_{ij} J_{ij}^{(k)} s_i s_j \right]} \right), \tag{4}$$

where we made explicit the dependence of the functions by the problem instance and the initial conditions, for clarity. In order to compute the minimum of H (denominator in (4) averaged over instances for large n, as customary in benchmarks we use Parisi formula's $\langle AR \rangle \simeq (-0.76 + 0.7 n^{-\frac{2}{3}}) n^{3/2}$ which is a very good estimate [14].

Figure 3 shows our overall results for the $\langle MSE \rangle$ as a function of t for different n, for instances for which we have trained the network as well as for unseen test instances. The results are consistent with the general illustrative considerations shown in Fig. 2 and show that for $n = 200$ DeepONet is clearly failing to simulate the ODEs since it is converging towards the expectation value for a random uniformly sampled trajectory $4/3$.

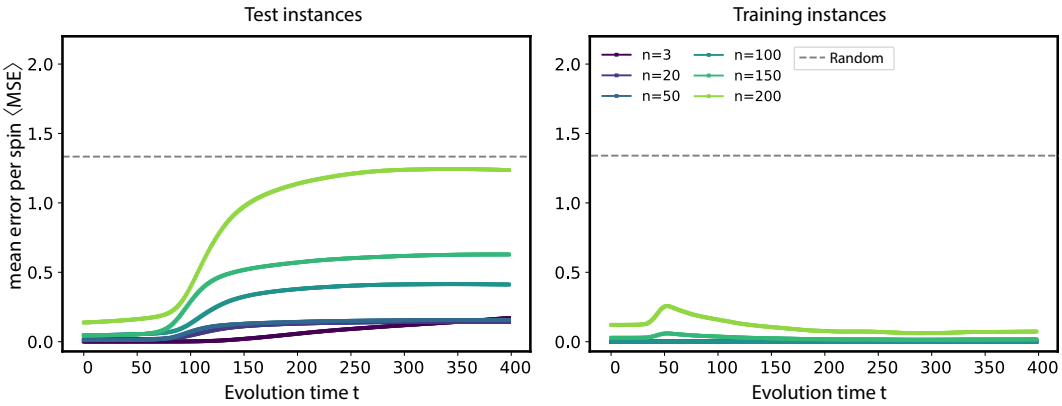

Figure 3: Instantaneous mean square error per spin over time as a function of problem size, for test (left) and training instances (right)

In Figure 4 we show the main results of our proof-of-concept, where we use the trained DeepONets in "optimization mode", evaluating their average performance versus the combinatorial solutions of the Ising problem provided by the exact ODEs. We observe that the $\langle AR \rangle$ of the neural version of the CIM is close in order of magnitude to the one obtained by the ODEs up to our very last benchmark size $n = 200$, for both the instance set and the test set. We also decided to benchmark the real-world use of optimization heuristics such as the CIM, for which the machine would be run multiple times (using different initial conditions) and only the best-found solutions would be returned. For this reason, we plot the average of the top 10% initial conditions for each problem instance using the dashed lines. For both best and average initialization, training loss is negligible. In the lower panel, we report respectively the runtimes of the neural networks when operated for inference (substantially flat at 2 milliseconds) and the time it took to train them via our procedures (linearly increasing up to our cutoff of about 1h of time).

[6]Usually the solution is computed at the final time $t = T$ but it is computationally efficient to scan for the best solution found across all evolution times, allowing for the rare statistical fluctuation when a higher quality solution is found in proximity of the bifurcation (see for instance 2 for n=100, the orange trajectory changes sign mid-way)

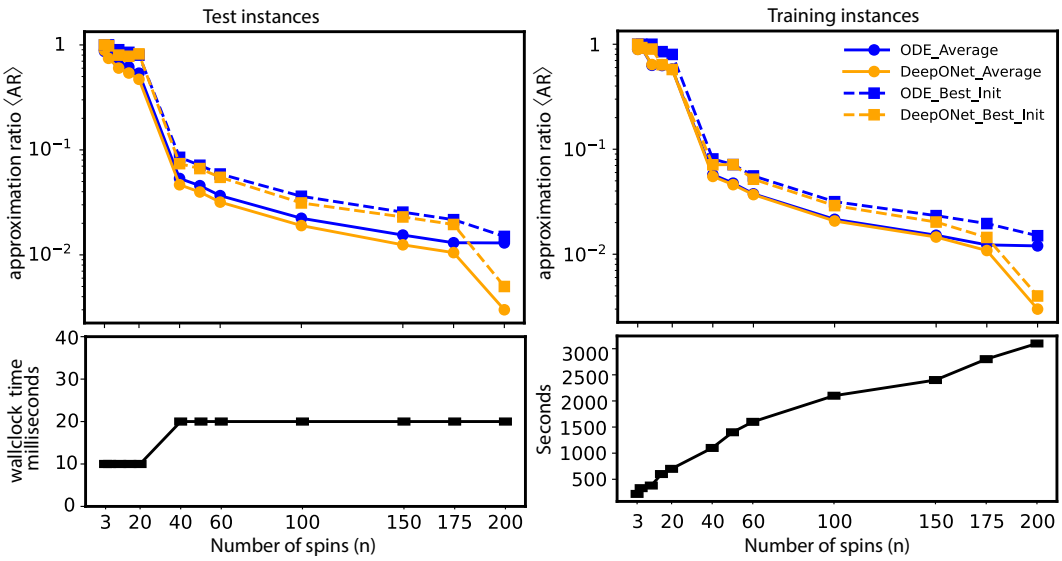

Figure 4: Comparative optimization performance results of DeepONets representation of CIM versus the exact integration, as a function of problem size. The left panel refers to statistics on test instances after training, while the right panel shows the statistics restricted to the instances used during training. Lower panels report on the times associated with the execution and the training of the networks used.

1.2 Discussion and Conclusions

In this workshop paper, we presented a proof-of-concept study showing that a DeepONet architecture is capable of efficiently simulating the coupled dynamics of ODEs - mapping the vectors of initial conditions x_0 and Ising coefficients J_{ij} to discrete spin configurations s_i obtained by projecting real functions $x_i(t)$ (solutions of the ODEs) on their sign. Our results show that after training, the network is capable of efficiently optimizing problems and initial conditions of the same size, up to and beyond 175 spins, which is a highly non-trivial task, especially in a regime with many coupled equations undergoing semi-chaotic dynamics.

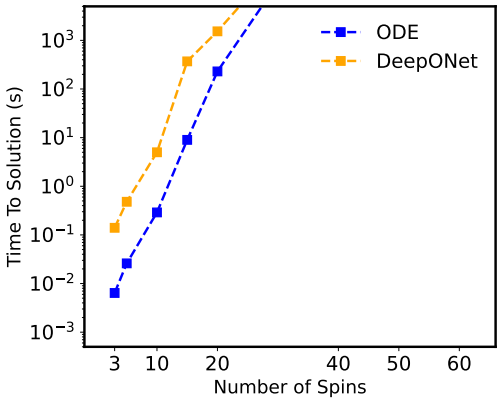

Figure 5: Average Time to Solution computed at P=99% for DeepONets representation of CIM versus the exact ODE integration, as a function of problem size. Probability of success has been computed using R_0=1000 and τ for each run was measured empirically as wall-clock time of the computer running the models.

Our tests have been limited by a rather small computational cut-off for training, and it is expected that our observations could generalize to higher n if sufficient data is allowed, and if the hyperparameters are scaled correctly. However, the dynamic described by Eq.(2) is unsophisticated if the purpose is to have a competitive Ising solver. Indeed the approximation ratios reported in Fig. 4 are rather unimpressive also for the ODE system, as it can be illustrated if we plot the common Time to solution ($\langle TTS \rangle$) metric as a function of the number of spins in Figure 5, where it is clear that the setup cannot reach the global optimum beyond 20 spins [7]. $\langle TTS \rangle$ is defined [15] as the mean time required to find the optimal solution of the problem at least one with a fixed probability P - which for independent runs of stochastic solvers is

[7] There is no immediate reason for which the absolute performance of the solver should correlate with the ability of the neural network to replicate the performance of the numerical integration.

$\langle TTS \rangle = \tau \log(1-P)/\log(1-R^\star/R_0)$ where R_0 is the number of i.i.d. runs, each of duration τ, and $R^\star$ is the number of those runs that end up in the global optimum.

State-of-art Ising Machines include more complex dynamics with more equations per binary variable, featuring a richer, more chaotic bifurcation phenomenology [3] and leveraging stochastic noise terms during the evolution [16, 17, 18]. An important task for proper benchmarking will be to learn optimal parameter setting strategies on-the-fly based on the instance to be solved [19], which is currently an open problem, that could be tackled in principle by neural training. Learning those models will be a challenge but will lead to more understanding on the power and limits of neural operators for the purpose of combinatorial optimization. Ultimately, a DeepONet could also be trained on a variety of different dynamical algorithms, including models that are not expressible via differential equations. The primary aim of this initial proof-of-concept is to assess how resilient and applicable our framework is in tackling a varied range of issues that will arise when we will apply the methods on state-of-art heuristic Ising Machine strategies.

2 Aknowledgements

This work has been funded by the National Science Foundation CCF-1918549. A.T. has been supported by the USRA Feynman Quantum Academy internship program. We thank Prof. Peter McMahon for insightful comments on our project.

References

[1] Naeimeh Mohseni, Peter L McMahon, and Tim Byrnes. Ising machines as hardware solvers of combinatorial optimization problems. *Nature Reviews Physics*, 4(6):363–379, 2022.

[2] Abhishek Kumar Singh, Kyle Jamieson, Peter L McMahon, and Davide Venturelli. Ising machines dynamics and regularization for near-optimal mimo detection. *IEEE Transactions on Wireless Communications*, 21(12):11080–11094, 2022.

[3] Y Yamamoto, T Leleu, S Ganguli, and H Mabuchi. Coherent ising machinesquantum optics and neural network perspectives. *Applied Physics Letters*, 117(16), 2020.

[4] Juntao Wang, Daniel Ebler, KY 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.

[5] Farhad Khosravi, Ugur Yildiz, Artur Scherer, and Pooya Ronagh. Non-convex quadratic programming using coherent optical networks. *arXiv preprint arXiv:2209.04415*, 2022.

[6] John J Hopfield and David W Tank. neural computation of decisions in optimization problems. *Biological cybernetics*, 52(3):141–152, 1985.

[7] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. *Advances in neural information processing systems*, 30, 2017.

[8] Martin JA Schuetz, J Kyle Brubaker, and Helmut G Katzgraber. Combinatorial optimization with physics-inspired graph neural networks. *Nature Machine Intelligence*, 4(4):367–377, 2022.

[9] Martin JA Schuetz, J Kyle Brubaker, Zhihuai Zhu, and Helmut G Katzgraber. Graph coloring with physics-inspired graph neural networks. *Physical Review Research*, 4(4):043131, 2022.

[10] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces. *arXiv preprint arXiv:2108.08481*, 2021.

[11] Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data. *Computer Methods in Applied Mechanics and Engineering*, 393:114778, 2022.

[12] Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

[13] Ryan Hamerly, Takahiro Inagaki, Peter L McMahon, Davide Venturelli, Alireza Marandi, Tatsuhiro Onodera, Edwin Ng, Carsten Langrock, Kensuke Inaba, Toshimori Honjo, et al. Experimental investigation of performance differences between coherent ising machines and a quantum annealer. *Science advances*, 5(5):eaau0823, 2019.

[14] Maxime Dupont, Bram Evert, Mark J Hodson, Bhuvanesh Sundar, Stephen Jeffrey, Yuki Yamaguchi, Dennis Feng, Filip B Maciejewski, Stuart Hadfield, M Sohaib Alam, et al. Quantum enhanced greedy solver for optimization problems. *arXiv preprint arXiv:2303.05509*, 2023.

[15] Troels F Rønnow, Zhihui Wang, Joshua Job, Sergio Boixo, Sergei V Isakov, David Wecker, John M Martinis, Daniel A Lidar, and Matthias Troyer. Defining and detecting quantum speedup. *science*, 345(6195):420–424, 2014.

[16] Sam Reifenstein, Satoshi Kako, Farad Khoyratee, Timothée Leleu, and Yoshihisa Yamamoto. Coherent ising machines with optical error correction circuits. *Advanced Quantum Technologies*, 4(11):2100077, 2021.

[17] Hayato Goto, Kotaro Endo, Masaru Suzuki, Yoshisato Sakai, Taro Kanao, Yohei Hamakawa, Ryo Hidaka, Masaya Yamasaki, and Kosuke Tatsumura. High-performance combinatorial optimization based on classical mechanics. *Science Advances*, 7(6):eabe7953, 2021.

[18] George Mourgias-Alexandris, Hitesh Ballani, Natalia G Berloff, James H Clegg, Daniel Cletheroe, Christos Gkantsidis, Istvan Haller, Vassily Lyutsarev, Francesca Parmigiani, Lucinda Pickup, et al. Analog iterative machine (aim): using light to solve quadratic optimization problems with mixed variables. *arXiv preprint arXiv:2304.12594*, 2023.

[19] Stochastic benchmark package - documentation. https://github.com/usra-riacs/stochastic-benchmark.