

Prediction and compression of lattice QCD data using machine learning algorithms on quantum annealer

Boram Yoon,^{a,*} Chia Cheng Chang,^{b,c,d} Garrett T. Kenyon,^e Nga T.T. Nguyen^e and Ermal Rrapaj^d

^a*CCS-7, Computer, Computational and Statistical Sciences Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA*

^b*RIKEN iTHEMS, Wako, Saitama 351-0198, Japan*

^c*Department of Physics, University of California, Berkeley, California 94720, USA*

^d*Nuclear Science Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA*

^e*CCS-3, Computer, Computational and Statistical Sciences Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA*

E-mail: boram@lanl.gov

We present regression and compression algorithms for lattice QCD data utilizing the efficient binary optimization ability of quantum annealers. In the regression algorithm, we encode the correlation between the input and output variables into a sparse coding machine learning algorithm. The trained correlation pattern is used to predict lattice QCD observables of unseen lattice configurations from other observables measured on the lattice. In the compression algorithm, we define a mapping from lattice QCD data of floating-point numbers to the binary coefficients that closely reconstruct the input data from a set of basis vectors. Since the reconstruction is not exact, the mapping defines a lossy compression, but, a reasonably small number of binary coefficients are able to reconstruct the input vector of lattice QCD data with the reconstruction error much smaller than the statistical fluctuation. In both applications, we use D-Wave quantum annealers to solve the NP-hard binary optimization problems of the machine learning algorithms.

*The 38th International Symposium on Lattice Field Theory, LATTICE2021 26th-30th July, 2021
Zoom/Gather@Massachusetts Institute of Technology*

*Speaker

1. Introduction

In lattice quantum chromodynamics (QCD) simulations, a large number of observables are measured on each Monte Carlo sample of the gauge field. The modern lattice QCD calculations targeting precise estimates often require the observable measurements computationally more expensive than the generation of the gauge field and produce $O(\text{PB})$ of data that need to be stored for a long term analysis. Since the measured observables share the same background lattice, however, their statistical fluctuations are correlated with each other. By exploiting the correlations, one can predict the values of the computationally expensive observables from the values of the computationally cheap observables, and compress the data for significantly reduced storage requirements.

Machine learning (ML) techniques provide powerful algorithms building models based on the correlation pattern of data. Various approaches enhancing the lattice QCD calculations using ML have been explored in recent studies [1–4]. However, the ML training, which is the procedure of finding optimal model parameters describing the data, of most of the ML algorithms involve nontrivial optimization problems. The optimization problems are computationally challenging as they are typically non-convex and the dimension of optimization parameters is very large. Stochastic approaches, such as the Adam optimizer [5], are used in many algorithms, but it demands expensive computations to obtain a near-optimal solution, especially for large models.

The optimization problems in ML algorithms can be solved using quantum annealers, such as the D-Wave system [6], which realizes the Ising spin system at a very low temperature. Starting from a transverse field Hamiltonian, the quantum processor of the D-Wave system performs quantum annealing towards the target Ising Hamiltonian. After the quantum annealing, the final state of the Ising spin system falls into a low-lying energy solution of the Ising Hamiltonian. Since the annealing time for each sample takes only $O(10)$ microseconds, a solution close to the ground-state can be obtained within a short wallclock time by repeating the quantum annealing and taking the smallest energy solution. Therefore, the optimization problem in ML algorithms can be solved using quantum annealing by mapping the optimization problem onto the structure of the Ising Hamiltonian:

$$H(\mathbf{h}, \mathbf{J}, \mathbf{s}) = \sum_i^{N_q} h_i s_i + \sum_{i<j}^{N_q} J_{ij} s_i s_j, \tag{1}$$

where $s_i = \pm 1$ is the binary spin variable, and h_i and J_{ij} are real-valued qubit biases and coupling strengths that can be set by a user. Note that this Ising model is isomorphic to a quadratic unconstrained binary optimization (QUBO) problem when defined in terms of $a_i = (s_i + 1)/2$.

Utilizing quantum annealing, in this work, we present a ML regression algorithm predicting the values of the lattice QCD observables [7] and a lossy data compression algorithm reducing the data storage requirement of the lattice QCD data [8].

2. Regression algorithm using quantum annealer

ML prediction of the unmeasured but expensive observables from the measured but cheap observables was first discussed in Ref. [1] and applied to a prediction of quasiparton distribution function matrix elements in Ref. [2]. The basic idea is the following. Assume that we have measurements on M independent lattice configurations. A set of common observables \mathbf{X} are

measured on all M configurations, while a target observable O is measured only on the first $N \ll M$ configurations. We call the first N configurations the *labeled dataset*, and the rest of the $M - N$ configurations the *unlabeled dataset*. Using the labeled dataset, one can train a ML regression model f to yield the prediction of O from \mathbf{X} for each configuration as $O_P^{(k)} = f(\mathbf{X}^{(k)})$, where k is the configuration index, and $O_P^{(k)}$ is the ML prediction of $O^{(k)}$. After training the correlation pattern between \mathbf{X} and O , the trained model can make predictions of $O^{(k)}$ for the unlabeled dataset.

Quantification of the prediction uncertainties is an important issue when applying ML to scientific data. All ML predictions may have prediction errors since the trained model is empirical, and the data used for the training have statistical noise. When estimating the expectation value of an observable through the Monte Carlo average, therefore, the simple average over the machine learning predictions $\frac{1}{M} \sum_k O_P^{(k)}$ could be a biased estimator of the target expectation value $\langle O \rangle$. For ML predictions on statistical data, however, an unbiased estimator can be defined following the bias correction algorithms developed in the truncated solver method [9] and the all-mode averaging [10]. In this approach, the labeled dataset is split into training (N_t) and bias-correction (N_b) datasets, where $N_t + N_b = N$, and the bias correction of the regression model trained on the training dataset can be carried out using the bias correction dataset as following:

$$O_P^{BC} = \frac{1}{M} \sum_{k=N_t+1}^{N+M} O_P^{(k)} + \frac{1}{N_b} \sum_{k=N_t+1}^N \left(O^{(k)} - O_P^{(k)} \right), \quad (2)$$

where O_P^{BC} is an unbiased estimator of the expectation value of O : $\langle O_P^{BC} \rangle = \langle O_P \rangle + \langle O - O_P \rangle = \langle O \rangle$. Here the second term is the bias correction term estimating the difference between the ground truth and its ML prediction. In the calculation of the statistical error of O_P^{BC} , the bias correction term naturally increases the size of the statistical error accounting for the prediction quality, so it provides a statistically sound uncertainty quantification of the ML predictions.

We design a new regression algorithm based on sparse coding [11], which is an unsupervised machine learning technique. For a set of input vectors of $\{\mathbf{X}^{(k)} \in \mathbb{R}^D\}_{k=1}^M$, sparse coding finds a matrix $\boldsymbol{\phi} \in \mathbb{R}^{D \times N_q}$, which is called the sparse coding dictionary, and the minimum number of nonzero coefficients $\{\mathbf{a}^{(k)} \in \mathbb{R}^{N_q}\}_{k=1}^M$ such that they reconstruct the input data as the following linear equation: $\mathbf{X}^{(k)} \approx \boldsymbol{\phi} \mathbf{a}^{(k)}$. This can be written as the following optimization problem:

$$\min_{\boldsymbol{\phi}} \sum_k \min_{\mathbf{a}^{(k)}} \left[\frac{1}{2} \|\mathbf{X}^{(k)} - \boldsymbol{\phi} \mathbf{a}^{(k)}\|^2 + \lambda \|\mathbf{a}^{(k)}\|_0 \right], \quad (3)$$

where λ is the sparsity penalty parameter. Here the difficult part is the optimization in $\mathbf{a}^{(k)}$ because the L_0 -norm in the sparsity penalty term makes the problem NP-hard [12].

The $\mathbf{a}^{(k)}$ -optimization part can be mapped onto the D-Wave through the following [13, 14]:

$$\mathbf{h} = -\boldsymbol{\phi}^T \mathbf{X} + \lambda + \frac{1}{2}, \quad \mathbf{J} = \boldsymbol{\phi}^T \boldsymbol{\phi}, \quad s = 2\mathbf{a} - 1. \quad (4)$$

In this way, one can solve the optimization in $\mathbf{a}^{(k)}$ using D-Wave, while the optimization for $\boldsymbol{\phi}$ can be done on classical computers. Note that, on the D-Wave, a_i corresponds to the spin up or down state, so it is restricted to a binary variable: $\mathbf{a}^{(k)} \in \{0, 1\}^{N_q}$.

Once trained, the $\boldsymbol{\phi}$ encodes the correlation pattern of the input data $\mathbf{X}^{(k)}$. Assume that we have a new vector \mathbf{X}' that has a few corrupted vector elements. If one calculates the sparse coefficients

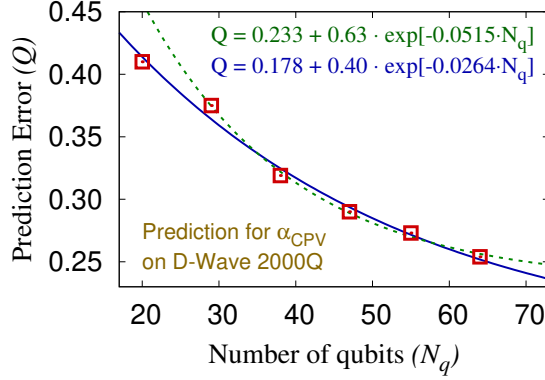


Figure 1: Prediction error of the CP-violating two-point correlation functions predicted using the sparse coding regression algorithm as a function of the number of qubits used for the prediction (red squares). An exponential ansatz is fitted to the data points for all N_q (blue line) and those excluding $N_q = 20$ (green line).

\mathbf{a}' of the \mathbf{X}' using the trained ϕ , it finds the most plausible basis vectors among ϕ describing the uncorrupted input vector elements. In the reconstruction $\phi\mathbf{a}'$, as a result, the corrupted input vector elements are replaced by those considered more natural based on the trained correlation pattern of ϕ . In the field of image processing, this procedure is called *inpainting* or *denoising* [15].

Using the inpainting feature of the sparse coding, a regression model can be constructed as follows. Consider N sets of labeled data $\{\mathbf{X}^{(i)}, y^{(i)}\}_{i=1}^N$, and M sets of the unlabeled data $\{\mathbf{X}^{(j)}\}_{j=N+1}^{N+M}$, where $\mathbf{X}^{(i)} \equiv \{x_1^{(i)}, x_2^{(i)}, \dots, x_D^{(i)}\}$ is the input vector (independent variable), and $y^{(i)}$ is the output variable (dependent variable). The goal is to build a regression model F using the labeled dataset to make predictions \hat{y} of y for an unseen input data \mathbf{X} as $F(\mathbf{X}) = \hat{y} \approx y$. The key idea is to concatenate the input and output variables of the labeled dataset into an extended vector, $\tilde{\mathbf{X}}^{(i)} \equiv \{x_1^{(i)}, x_2^{(i)}, \dots, x_D^{(i)}, y^{(i)}\}$, and calculate the dictionary matrix $\tilde{\phi} \in \mathbb{R}^{(D+1) \times N_q}$ of the extended vectors $\{\tilde{\mathbf{X}}^{(i)}\}_{i=1}^N$. For the prediction on the unlabeled dataset, build vectors $\tilde{\mathbf{X}}_o^{(j)} \equiv \{x_1^{(j)}, x_2^{(j)}, \dots, x_D^{(j)}, \bar{y}^{(j)}\}$, where $\bar{y}^{(j)}$ is an initial guess of $y^{(j)}$ set by the average value of $y^{(i)}$ in the labeled dataset. Then, using the dictionary $\tilde{\phi}$ obtained from the labeled dataset, find sparse representation $\mathbf{a}^{(j)}$ for $\tilde{\mathbf{X}}_o^{(j)}$ and calculate reconstruction as $\tilde{\mathbf{X}}'^{(j)} = \tilde{\phi}\mathbf{a}^{(j)}$. This replaces the outlier components, including $\bar{y}^{(j)}$, in $\tilde{\mathbf{X}}_o^{(j)}$ by the values that can be described by $\tilde{\phi}$. Finally, the prediction of $y^{(j)}$ is given by $\hat{y}^{(j)} = [\tilde{\mathbf{X}}'^{(j)}]_{D+1}$. A detailed procedure, including the pre-training and normalization improving the prediction performance, is described in Ref. [7].

To demonstrate the regression ability of the proposed algorithm, we apply it to the lattice QCD simulation data generated for the calculation of the charge-parity (CP) symmetry violating phase α_{CPV} of the neutron [16]. From the real and imaginary parts of the normal and pseudoscalar-projected two-point correlation functions calculated without CPV interactions at $t/a = 8, 9, \dots, 12$, we predict the imaginary part of the two-point correlation functions calculated in presence of the CP-violating interactions. In this example, the input vector has 20 variables and the output is a single variable. For this test, we train $\tilde{\phi}$ using 6976 data samples and test its prediction ability on 8640 data samples measured in Refs. [17, 18]. For the training and prediction, the \mathbf{a} -optimization is carried out using the D-Wave 2000Q quantum annealer, and the ϕ -optimization is performed using the stochastic gradient descent method on a classical computer. Because of the restricted

number of fully connected logical qubits on the D-Wave 2000Q hardware, the maximum dimension of the sparse representation $\mathbf{a}^{(k)}$ is limited to 64. Results presented in Fig. 1 show that the current prediction performance is limited by the maximum number of qubits of the quantum annealer. A detailed description of the quantum annealing and regression results are presented in Ref. [7].

3. Lossy compression algorithm using quantum annealer

Another interesting use of quantum annealers is data compression. Lattice QCD simulations produce a huge amount of data, but they are correlated with each other, so one can reduce the storage space requirement by exploiting the correlation. In most lattice QCD data analyses, furthermore, data precision sufficiently smaller than the statistical fluctuation is good enough, so lossy data compression is considered as a viable approach.

A lossy compression algorithm can be constructed in a similar structure as the sparse coding described in the previous section. Assume that we have a matrix $\boldsymbol{\phi} \in \mathbb{R}^{D \times N_q}$ and binary coefficients $\mathbf{a}^{(k)} \in \{0, 1\}^{N_q}$ that precisely reconstruct the input vectors $\mathbf{X}^{(k)} \in \mathbb{R}^D$ such that $\mathbf{X}^{(k)} \approx \boldsymbol{\phi} \mathbf{a}^{(k)}$ for all data index $k = 1, 2, 3, \dots, N$. The procedure defines a mapping from \mathbf{X} -space to \mathbf{a} -space:

$$\left\{ \mathbf{X}^{(k)} \mid \mathbf{X}^{(k)} \in \mathbb{R}^D \right\}_{k=1}^N \longrightarrow \left(\left\{ \mathbf{a}^{(k)} \mid \mathbf{a}^{(k)} \in \{0, 1\}^{N_q} \right\}_{k=1}^N, \boldsymbol{\phi} \in \mathbb{R}^{D \times N_q} \right). \quad (5)$$

Note that here we restrict the $a_i^{(k)}$ to a binary variable so that it can be stored in a single bit. Also, we focus on the cases where $N_q \ll N$, which is the typical situation in high-statistics data, so that the memory usage of $\boldsymbol{\phi}$ is sufficiently smaller than the size of the original data. This transformation forms a data compression since the data in \mathbf{a} -space require less memory than those in \mathbf{X} -space.

Such $\{\mathbf{a}^{(k)}\}$ and $\boldsymbol{\phi}$ can be obtained by minimizing the mean square reconstruction error,

$$\min_{\boldsymbol{\phi}} \sum_k \min_{\mathbf{a}^{(k)}} \left[\sum_{i=1}^D \left(X_i^{(k)} - [\boldsymbol{\phi} \mathbf{a}^{(k)}]_i \right)^2 \right]. \quad (6)$$

In general, a weight factor of inverse variance, $1/\sigma_{X_i}^2$, should be multiplied to the least-squares loss function to avoid the algorithm focusing on the large-variance components of the input vector. In this study, we assume that the data is standardized before the compression, so $1/\sigma_{X_i}^2 = 1$. Here, the $\boldsymbol{\phi}$ -optimization part can be performed on classical computers using stochastic gradient descent, but finding the optimal solution of the \mathbf{a} -vectors is an NP-hard problem because it is a binary optimization. However, the \mathbf{a} -optimization can be solved using the D-Wave quantum annealer after the transformation to the Ising Hamiltonian in Eq. (1) following

$$\mathbf{J} = 2\boldsymbol{\phi}^T \boldsymbol{\phi}, \quad h_i = -2 [\boldsymbol{\phi}^T \mathbf{X}]_i + [\boldsymbol{\phi}^T \boldsymbol{\phi}]_i, \quad s = 2\mathbf{a} - 1. \quad (7)$$

Since this is a lossy compression algorithm, the reconstruction is not exactly the same as the original data, and there could be reconstruction errors that introduce bias in the data analysis. However, the bias can be removed using a small portion of the original data following the bias correction approach given in Eq. (2). A detailed description can be found in Ref. [8].

To show the compression performance, we apply the algorithm to compress the vector and axial-vector nucleon three-point correlator data of the 16 timeslices calculated in Ref. [19]. Fig. 2

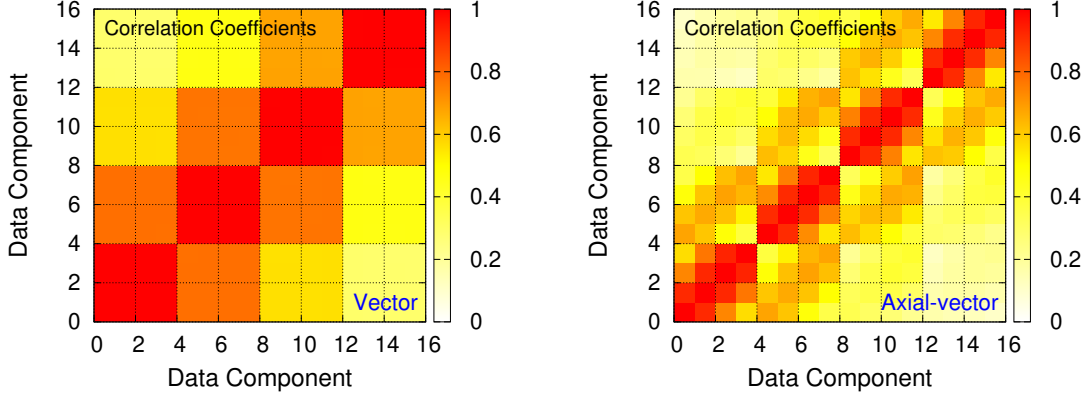


Figure 2: Correlation pattern of the 16 components of the vector (left) and the axial-vector (right) data. Red indicates the high correlation (correlation coefficient = 1), and white indicates no correlation.

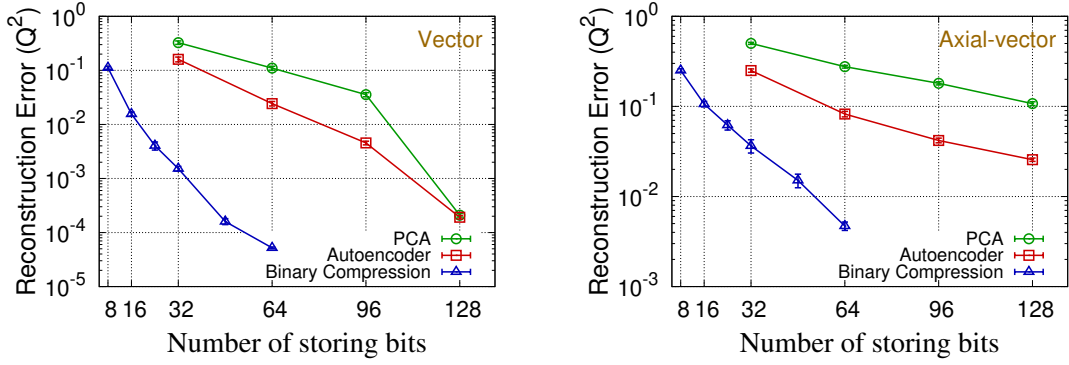


Figure 3: Squared reconstruction error for different number of storing bits. For the principal component analysis (PCA) and autoencoder (AE) approaches, the number of storing bits is calculated by $32 \times N_z$, assuming single-precision floating-point numbers.

shows their correlation pattern. To avoid the effects from the integrated control errors (ICE) [20] of the D-Wave hardware, we first carry out the compression tests using D-Wave’s simulated annealing sampler implemented in the D-Wave’s Ocean library [21], on classical computers. Fig. 3 shows the reconstruction error of the new algorithm in blue data points, and those of the other compression algorithms based on principal component analysis (PCA) and neural-network autoencoder for various different numbers of storing bits. The new binary compression algorithm shows a much smaller reconstruction error than others. Also, by comparing the vector and axial-vector results, one can see that the reconstruction error of the vector data is much smaller than the reconstruction error of the axial-vector data, because of the higher correlation in the vector channel. We also carry out the compression test using D-Wave 2000Q and Advantage systems but found slightly worse compression performance than the simulated annealing due to the ICE.

The compression can be used for outlier detection for rare events or data corruption, because a large reconstruction error for an input data vector indicates that the correlation pattern of the data is different from those used in the training of ϕ . Also, the relation between the original vector \mathbf{X} and the binary coefficients \mathbf{a} is linear, so one can replace the expensive operations on the floating-point

numbers of \mathbf{X} by the cheap operations on the single-bit coefficients \mathbf{a} .

4. Conclusion

We presented a new ML regression algorithm utilizing D-Wave quantum annealer as an efficient optimization algorithm finding optimal ML model parameters. Motivated by the inpainting feature of the sparse coding, a correlation pattern between the dependent and independent variables is trained to make predictions on unlabeled dataset. For the training and predictions, the D-Wave 2000Q quantum annealer is used to find the sparse coefficients of the sparse coding ML algorithm. The regression algorithm is applied to predict the two-point correlators of the CP-violating interactions. Results show that the regression performance is limited by the number of fully connected logical qubits of current D-Wave hardware, but promising on the future quantum annealing architectures.

We also presented a new lossy data compression algorithm based on ML using a D-Wave quantum annealer. The algorithm finds a set of basis vectors and their binary coefficients that precisely reconstruct the original data. Finding such binary coefficients is an NP-hard optimization problem, but the algorithm is formulated so that they can be obtained using the D-Wave quantum annealer with a short wallclock time. The compression algorithm applied to nucleon three-point correlator data shows a much better compression performance than the compression algorithms based on PCA or the autoencoder. Currently, the compression performance with D-Wave quantum annealer is limited by the ICE of the D-Wave hardware. On future architectures with improved ICE, however, the algorithm is expected to perform much better.

Acknowledgments

The QUBO optimizations were carried out using the D-Wave 2000Q at Los Alamos National Laboratory (LANL) and the D-Wave's Leap Quantum Cloud Service. Simulation data used for the numerical experiment were generated using the computer facilities at NERSC, OLCF, USQCD Collaboration, and institutional computing at LANL. This work was supported by the U.S. Department of Energy (DOE), Office of Science, Office of High Energy Physics under Contract No. 89233218CNA000001 and LANL LDRD program. Lawrence Berkeley National Laboratory (LBNL) is operated by The Regents of the University of California (UC) for the U.S. DOE under Federal Prime Agreement DE-AC02-05CH11231. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics, Quantum Horizons: QIS Research and Innovation for Nuclear Science under Award Number FWP-NQISCCAWL (CCC). ER acknowledges the NSF N3AS Physics Frontier Center, NSF Grant No. PHY-2020275, and the Heising-Simons Foundation (2017-228).

References

- [1] B. Yoon, T. Bhattacharya, and R. Gupta, *Machine Learning Estimators for Lattice QCD Observables*, *Phys. Rev. D* **100** (2019), no. 1 014504, [[arXiv:1807.05971](https://arxiv.org/abs/1807.05971)].

- [2] R. Zhang, Z. Fan, R. Li, H.-W. Lin, and B. Yoon, *Machine-learning prediction for quasiparton distribution function matrix elements*, *Phys. Rev. D* **101** (2020), no. 3 034516, [arXiv:1909.10990].
- [3] G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racanière, D. J. Rezende, and P. E. Shanahan, *Equivariant flow-based sampling for lattice gauge theory*, *Phys. Rev. Lett.* **125** (2020), no. 12 121601, [arXiv:2003.06413].
- [4] S. Bulusu, M. Favoni, A. Ipp, D. I. Müller, and D. Schuh, *Generalization capabilities of translationally equivariant neural networks*, *Phys. Rev. D* **104** (2021), no. 7 074504, [arXiv:2103.14686].
- [5] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017.
- [6] “D-Wave System Documentation.”
<https://docs.dwavesys.com/docs/latest/index.html>. Accessed: 2021-05-10.
- [7] N. T. Nguyen, G. T. Kenyon, and B. Yoon, *A regression algorithm for accelerated lattice QCD that exploits sparse inference on the D-Wave quantum annealer*, *Sci. Rep.* **10** (2020), no. 1 10915, [arXiv:1911.06267].
- [8] B. Yoon, N. T. T. Nguyen, C. C. Chang, and E. Rrapaj, *Lossy compression of statistical data using quantum annealer*, arXiv:2110.02142.
- [9] G. S. Bali, S. Collins, and A. Schafer, *Effective noise reduction techniques for disconnected loops in Lattice QCD*, *Comput. Phys. Commun.* **181** (2010) 1570–1583, [arXiv:0910.3970].
- [10] T. Blum, T. Izubuchi, and E. Shintani, *New class of variance-reduction techniques using lattice symmetries*, *Phys. Rev.* **D88** (2013), no. 9 094503, [arXiv:1208.4349].
- [11] C. Rozell, D. Johnson, R. Baraniuk, and B. Olshausen, *Sparse coding via thresholding and local competition in neural circuits*, *Neural Computation* **20** (October, 2008) 2526–2563.
- [12] B. K. Natarajan, *Sparse approximate solutions to linear systems*, *SIAM J. Comput.* **24** (Apr., 1995) 227–234.
- [13] N. T. T. Nguyen and G. T. Kenyon, *Solving sparse representation for object classification using quantum d-wave 2x machine*, in *The First IEEE International Workshop on Post Moore’s Era Supercomputing, PMES*, pp. 43–44, 2016.
- [14] N. T. T. Nguyen and G. T. Kenyon, *Image classification using quantum inference on the d-wave 2x*, in *2018 IEEE International Conference on Rebooting Computing (ICRC)*, pp. 1–7, 2018. arXiv:1905.13215.
- [15] B. Shen, W. Hu, Y. Zhang, and Y.-J. Zhang, *Image inpainting via sparse representation*, in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 697–700, 2009.

- [16] B. Yoon, T. Bhattacharya, and R. Gupta, *Neutron Electric Dipole Moment on the Lattice*, *EPJ Web Conf.* **175** (2018) 01014, [[arXiv:1712.08557](https://arxiv.org/abs/1712.08557)].
- [17] T. Bhattacharya, V. Cirigliano, R. Gupta, E. Mereghetti, and B. Yoon, *Neutron Electric Dipole Moment from quark Chromoelectric Dipole Moment*, *PoS LATTICE2015* (2016) 238, [[arXiv:1601.02264](https://arxiv.org/abs/1601.02264)].
- [18] T. Bhattacharya, V. Cirigliano, R. Gupta, and B. Yoon, *Quark Chromoelectric Dipole Moment Contribution to the Neutron Electric Dipole Moment*, *PoS LATTICE2016* (2016) 225, [[arXiv:1612.08438](https://arxiv.org/abs/1612.08438)].
- [19] **PNDME** Collaboration, R. Gupta, T. Bhattacharya, V. Cirigliano, B. Yoon, Y.-C. Jang, and H.-W. Lin, *Isvector and flavor diagonal charges of the nucleon from 2+1+1 flavor QCD*, *PoS LATTICE2018* (2018) 131, [[arXiv:1812.03573](https://arxiv.org/abs/1812.03573)].
- [20] “ICE: Dynamic Ranges in h and J Values.” https://docs.dwavesys.com/docs/latest/c_qpu_1.html. Accessed: 2021-05-10.
- [21] “D-Wave Ocean Software.” <https://docs.ocean.dwavesys.com/>. Accessed: 2021-05-10.