scientific reports

Check for updates

OPEN

# Lossy compression of statistical data using quantum annealer

Boram Yoon[1]✉, Nga T. T. Nguyen[2], Chia Cheng Chang[3,4,5,6] & Ermal Rrapaj[4]

**We present a new lossy compression algorithm for statistical floating-point data through a representation learning with binary variables. The algorithm finds a set of basis vectors and their binary coefficients that precisely reconstruct the original data. The optimization for the basis vectors is performed classically, while binary coefficients are retrieved through both simulated and quantum annealing for comparison. A bias correction procedure is also presented to estimate and eliminate the error and bias introduced from the inexact reconstruction of the lossy compression for statistical data analyses. The compression algorithm is demonstrated on two different datasets of lattice quantum chromodynamics simulations. The results obtained using simulated annealing show 3–3.5 times better compression performance than the algorithm based on neural-network autoencoder. Calculations using quantum annealing also show promising results, but performance is limited by the integrated control error of the quantum processing unit, which yields large uncertainties in the biases and coupling parameters. Hardware comparison is further studied between the previous generation D-Wave 2000Q and the current D-Wave Advantage system. Our study shows that the Advantage system is more likely to obtain low-energy solutions for the problems than the 2000Q.**

Today's scientific computing and experiments often produce petabytes of floating-point data that need to be stored for post-processing or transferred to different computing centers. For example, modern lattice quantum chromodynamics (QCD) simulations targeting accurate precision generate $O$(PB) of data[1,2] and store the data on storage systems for long-term analysis. In many applications, only a few significant figures of the stored data are required for the analysis, so lossy data compression algorithms are considered as viable approaches to reducing the data storage requirement and increasing the effective bandwidth for data movement.

Various lossy data compression algorithms recently proposed for floating-point arrays of scientific data include ISABELA[3], ZFP[4], SZ[5–7], and NUMARCK[8]. ISABELA provides in-situ compression based on interpolation using B-splines[9] after sorting multidimensional scientific data. ZFP uses block transformation for decorrelation and bit-plane encoding for a fixed-rate lossy compression. SZ is an error-bounded lossy compression algorithm based on fitting and predicting the successive data points. NUMARCK achieves the data compression by approximating the temporal changes using the K-means data clustering algorithm[10].

For statistical data, it is possible to detect the correlation pattern of the data components using machine learning techniques and exploit the learned correlation for efficient lossy data compression. An example is the approaches based on the autoencoder[11–13]. Unsupervised machine learning techniques allow us to find efficient codings of the input data. They work as data compression algorithms since the coding is typically a lower-dimensional representation of the original data. The compression performance of the representation learning can be maximized by restricting the codes to be binary variables so that each code can be stored in a bit. However, an encoder with binary codes generally involves binary optimization for finding the optimal codes, which is an NP-hard problem.

Quantum annealing is an approach to solving such optimization problems using adiabatic quantum computation (AQC). In contrast to the gate model quantum computation[14,15], AQC performs an adiabatic time evolution from a ground state of a simple initial Hamiltonian to the state of the final Hamiltonian of the target problem[16–18]. When the Hamiltonian is evolved sufficiently slowly, the adiabatic theorem guarantees that the state remains in the ground state of the instantaneous Hamiltonian. Quantum annealing is AQC with a relaxation of the adiabatic condition in an open system at finite temperature[19]. It solves combinatorial optimization problems using the quantum effect tunneling through barriers between local minima[20–23]. Although its ability to demonstrate the

[1]CCS-7, Computer, Computational and Statistical Sciences Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA. [2]CCS-3, Computer, Computational and Statistical Sciences Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA. [3]RIKEN iTHEMS, Wako, Saitama 351-0198, Japan. [4]Department of Physics, University of California, Berkeley, CA 94720, USA. [5]Nuclear Science Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA. [6]LinkedIn Corporation, Sunnyvale, CA 94085, USA. ✉email: boram@lanl.gov

quantum speedup over classical computation is still controversial[24], quantum annealing has been successfully applied to various NP-hard and NP-complete problems[25–29].

The quantum processor of the D-Wave systems realizes the quantum annealing to find low-lying energy states of the target Ising Hamiltonian starting from the transverse field Hamiltonian[30]. In general, the quantum annealing will require an exponentially large number of samples in order to recover the most optimal solution to large binary optimization problems. However, the annealing time for each sample takes only $O(10)$ microseconds. Furthermore, we show in this work that low-energy solutions, which can be obtained from only a small number of samples, are sufficient for our proposed sparse coding compression algorithm.

In this paper, we propose a new data compression algorithm based on a representation learning. For a maximum compression, we use binary codes, and formulate the problem in a quadratic unconstrained binary optimization (QUBO) form so that it can be solved on Ising solvers such as the D-Wave quantum annealer, as described in "Method" section. As a result, the algorithm guarantees the compression ratio while optimizing for the smallest loss. We also present a bias correction procedure that removes the bias and estimate the errors due to the inexact reconstruction from the lossy-compressed statistical data. In "Numerical experiments" section, the proposed compression algorithm is demonstrated for two different lattice QCD datasets using simulated annealing, and D-Wave's 2000Q and the Advantage System.

## Method

**Data compression.** The goal of this algorithm is to find a matrix $\boldsymbol{\phi} \in \mathbb{R}^{D\times N_q}$ and binary coefficients $\boldsymbol{a}^{(k)} \in \{0,1\}^{N_q}$ that precisely reconstruct the input vectors $\mathbf{X}^{(k)} \in \mathbb{R}^D$ such that $\mathbf{X}^{(k)} \approx \boldsymbol{\phi}\boldsymbol{a}^{(k)}$ for all data index $k = 1, 2, 3, \ldots, N$. The procedure defines a mapping from $\mathbf{X}$-space to $\boldsymbol{a}$-space:

$$\left\{\mathbf{X}^{(k)} | \mathbf{X}^{(k)} \in \mathbb{R}^D, k = 1, 2, \ldots, N\right\} \longrightarrow \left(\left\{\boldsymbol{a}^{(k)} | \boldsymbol{a}^{(k)} \in \{0,1\}^{N_q}, k = 1, 2, \ldots, N\right\}, \boldsymbol{\phi} \in \mathbb{R}^{D\times N_q}\right). \tag{1}$$

Here the coefficients in the $\boldsymbol{a}$-space are restricted to binary variables so that it can be stored in a single bit. Additionally, we restrict $N_q \ll N$ so that the memory usage of $\boldsymbol{\phi}$ is comparatively small to the uncompressed data, which for high-statistics datasets where compression is necessary is of $N \gtrsim \mathrm{O}(10^4)$. As a result, the data in $\boldsymbol{a}$ -space uses less memory space than those in $\mathbf{X}$-space, and results in data compression.

One possible solution of the mapping ($\{\boldsymbol{a}^{(k)}\}$ and $\boldsymbol{\phi}$) can be obtained by minimizing the mean square error of the reconstruction as following:

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

When the underlying data exhibits heteroskedasticity, a weight factor of inverse variance, $1/\sigma_{X_i}^2$, needs to be multiplied to each term of the least-squares loss function to avoid the algorithm focusing on the reconstruction of the large-variance components of the input vector and to make the reconstruction error uniform. The same effect can be achieved by standardizing the input data $\mathbf{X}$ in the data preparation. The resulting optimization problem is mapped to a QUBO

$$H(\boldsymbol{h}, J, \boldsymbol{s}) = \sum_i^{N_q} h_i s_i + \sum_{i<j}^{N_q} J_{ij} s_i s_j, \tag{3}$$

through the transformation given below:

$$J = 2\boldsymbol{\phi}^T\boldsymbol{\phi}, \qquad h_i = -2\left[\boldsymbol{\phi}^T\mathbf{X}\right]_i + \left[\boldsymbol{\phi}^T\boldsymbol{\phi}\right]_i, \qquad \boldsymbol{s} = 2\boldsymbol{a} - 1. \tag{4}$$

Note that structure of the transformation is similar to the one used in sparse coding[31–33], but the compression algorithm does not require the constraints $\left[\boldsymbol{\phi}^T\boldsymbol{\phi}\right]_i = 1$, placed in the sparse coding.

After obtaining the solution of $\boldsymbol{a}^{(k)}$ for a given $\boldsymbol{\phi}$, we update $\boldsymbol{\phi}$ using stochastic gradient decent on a classical computer. The optimizations for $\boldsymbol{a}^{(k)}$ and $\boldsymbol{\phi}$ are iterated until they reach to a stationary solution. The procedure can be summarized as following.

(1) Initialize $\{\boldsymbol{a}^{(k)}\}$ and $\boldsymbol{\phi}$ with random numbers or initial guesses.
(2) Take a random mini-batch of size $N_b$ from the $N$ samples of $\mathbf{X}^{(k)}$.
(3) Within the mini-batch, fix $\boldsymbol{\phi}$ and find $\{\boldsymbol{a}^{(k)}\}$ that minimizes Eq. (2).
(4) Within the mini-batch, fix $\{\boldsymbol{a}^{(k)}\}$ and update $\boldsymbol{\phi}$ towards the optimum solution of Eq. (2) with a learning rate $\eta$.
(5) Repeat (2)–(4) until it reaches the minimum reconstruction error.

Here the mini-batch size $N_b$ and the learning rate $\eta$ control the convergence of the algorithm.

**Bias correction.** In many scientific applications, such as the Monte Carlo simulations, our major concern is the expectation value of a function of the statistical variables $\langle f(\mathbf{X})\rangle$. With the samples $\mathbf{X}^{(k)}$, the expectation value is usually estimated by a simple average over $k$. When using the compressed data in $\boldsymbol{a}$-space, however, the lossy-compression introduces reconstruction error $\mathbf{X}^{(k)} \neq \boldsymbol{\phi}\boldsymbol{a}^{(k)} \equiv \tilde{\mathbf{X}}^{(k)}$. As a result, a simple average $\frac{1}{N}\sum_{k=1}^{N} f(\tilde{\mathbf{X}}^{(k)})$ as an estimator of $\langle f(\mathbf{X})\rangle$ is biased.

An unbiased estimator $\bar{O}^{\mathrm{BC}}$ can be defined by using a small portion of the original data $\mathbf{X}^{(k)}$:

$$\bar{O}^{\mathrm{BC}} = \frac{1}{N}\sum_{k=1}^{N} f(\tilde{\mathbf{X}}^{(k)}) + \frac{1}{N_{\mathrm{bc}}}\sum_{k=1}^{N_{\mathrm{bc}}}\left(f(\mathbf{X}^{(k)}) - f(\tilde{\mathbf{X}}^{(k)})\right). \tag{5}$$

Here the first term on the right hand side is a sloppy estimator of $\langle f(\mathbf{X})\rangle$, and the second term is a bias correction term that makes the estimator satisfy $\langle \bar{O}^{\mathrm{BC}}\rangle = \langle f(\mathbf{X})\rangle$. Note that in the second term, we use the first $N_{\mathrm{bc}}$ samples out of total $N$ samples as a bias correction dataset, assuming the data samples are independent and identically distributed. Depending on the data characteristics, however, one could take the maximally separated or randomly chosen $N_{\mathrm{bc}}$ samples for the bias correction dataset.

In addition to the $\{\boldsymbol{a}^{(k)}\}$ and $\boldsymbol{\phi}$, for a bias correction in the reconstruction, one needs to store the $N_{\mathrm{bc}}$ samples of the original data $\{\mathbf{X}^{(k)}|k = 1, 2, \ldots, N_{\mathrm{bc}}\}$. As explained in "Quality indicator for lossy-compression" section, the statistical error of $\bar{O}^{\mathrm{BC}}$ induced by the bias correction term depends on $N_{\mathrm{bc}}$ and the correlation between $f(\mathbf{X}^{(k)})$ and $f(\tilde{\mathbf{X}}^{(k)})$. For a good compression, which yields high correlation between correlation between $f(\mathbf{X}^{(k)})$ and $f(\tilde{\mathbf{X}}^{(k)})$, the bias $f(\mathbf{X}) - f(\tilde{\mathbf{X}}^{(k)})$ can be estimated precisely from a small number of samples, so one can take $N_{\mathrm{bc}} \ll N$. A similar structure of bias correction has been demonstrated in the machine learning regressions on statistical data[34,35].

In the calculation of the statistical error of $\bar{O}^{\mathrm{BC}}$, the correlation between the sloppy estimator and the bias correction term should be taken into account. One approach to make the procedure simple is binning the data so that each bin has a certain number of bias correction data samples and the data in different bins are uncorrelated with each other. Assuming that the number of bins $N_{\mathrm{bin}}$ divides $N$ and $N_{\mathrm{bc}}$, Eq. (5) can be rewritten as

$$\bar{O}^{\mathrm{BC}} = \frac{1}{N_{\mathrm{bin}}}\sum_{i=1}^{N_{\mathrm{bin}}}\left[\frac{1}{M}\sum_{k=iM+1}^{(i+1)M} f(\tilde{\mathbf{X}}^{(k)}) + \frac{1}{M_{\mathrm{bc}}}\sum_{k=iM+1}^{iM+M_{\mathrm{bc}}}\left(f(\mathbf{X}^{(k)}) - f(\tilde{\mathbf{X}}^{(k)})\right)\right] \tag{6}$$

$$\equiv \frac{1}{N_{\mathrm{bin}}}\sum_{i=1}^{N_{\mathrm{bin}}} \bar{O}_i^{\mathrm{BC},b}, \tag{7}$$

where $M = N/N_{\mathrm{bin}}$ and $M_{\mathrm{bc}} = N_{\mathrm{bc}}/N_{\mathrm{bin}}$. In this rearrangement, the statistical error of $\bar{O}^{\mathrm{BC}}$ can be calculated by $\sigma_{\bar{O}^{\mathrm{BC}}} = \sigma_{\bar{O}^{\mathrm{BC},b}}/\sqrt{N_{\mathrm{bin}}}$. Again, note that the first $M_{\mathrm{bc}}$ samples in each bin are used for the bias correction, but one can take maximally separated or randomly chosen samples for the bias correction dataset, depending on the characteristics of the data.

**Quality indicator for lossy-compression.** To measure the quality of lossy-compression on statistical data, we define the $Q^2$ as

$$Q^2 \equiv \frac{1}{D}\sum_{i=1}^{D}\frac{\sigma^2_{X_i-\tilde{X}_i}}{\sigma^2_{\tilde{X}_i}}, \tag{8}$$

where $\sigma^2_{X_i-\tilde{X}_i}$ is the variance of $X_i - \tilde{X}_i$. This parameter is an indicator of the statistical error increase due to the lossy-compression after the bias correction as following. Consider a simple bias-corrected average of independent observables

$$\bar{\mathbf{X}}^{\mathrm{BC}} = \frac{1}{N}\sum_{k=1}^{N}\tilde{\mathbf{X}}^{(k)} + \frac{1}{N_{\mathrm{bc}}}\sum_{k=1}^{N_{\mathrm{bc}}}\left(\mathbf{X}^{(k)} - \tilde{\mathbf{X}}^{(k)}\right). \tag{9}$$

The variance of the $i$-th component of $\bar{\mathbf{X}}$ can be approximated as

$$\sigma^2_{\bar{X}_i^{\mathrm{BC}}} \approx \frac{1}{N}\sigma^2_{\tilde{X}_i} + \frac{1}{N_{\mathrm{bc}}}\sigma^2_{X_i-\tilde{X}_i} \tag{10}$$

$$\approx \frac{\sigma^2_{\tilde{X}_i}}{N}\left(1 + \frac{N}{N_{\mathrm{bc}}}\frac{\sigma^2_{X_i-\tilde{X}_i}}{\sigma^2_{\tilde{X}_i}}\right), \tag{11}$$

where the first approximation assumes a small correlation between the two terms in Eq. (9), and the second approximation assumes a good lossy-compression that gives $\sigma^2_{\tilde{X}_i} \approx \sigma^2_{\bar{X}_i}$. Assuming a small reconstruction error satisfying $\sigma^2_{X_i-\tilde{X}_i}/\sigma^2_{\bar{X}_i} \ll N/N_{\mathrm{bc}}$, the expected statistical error increase due to the bias correction can be estimated as

$$\frac{\sigma_{\bar{X}_i^{\mathrm{BC}}}}{\sigma_{\bar{X}_i}} \approx 1 + \alpha\frac{N}{2N_{\mathrm{bc}}}\frac{\sigma^2_{X_i-\tilde{X}_i}}{\sigma^2_{\bar{X}_i}}, \tag{12}$$

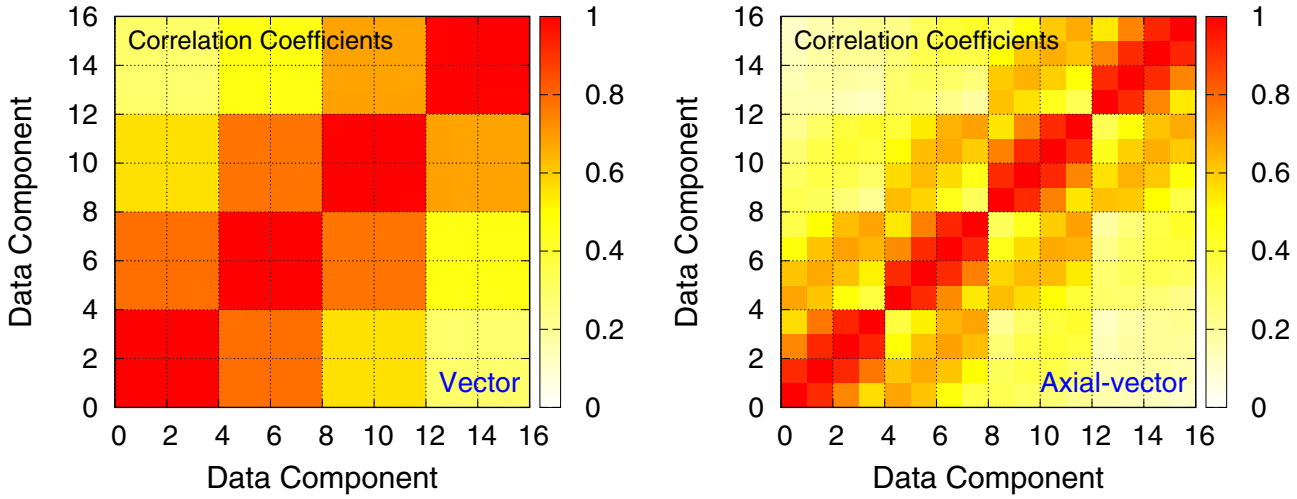



**Figure 1.** Correlation pattern of the 16 components of the vector (left) and the axial-vector (right) data. Red indicates the high correlation (correlation coefficient = 1), and white indicates no correlation.

where $\alpha = 1$ and $\sigma^2_{\tilde{X}_i} = \sigma^2_{X_i}/N$. It shows that the increase of the statistical error compared to that of the original data is proportional to ratio of the number of bias correction data, $N/N_{\rm bc}$, and the normalized variance of the reconstruction error, $\sigma^2_{X_i-\tilde{X}_i}/\sigma^2_{X_i}$. Hence, we define the quality of the compression by taking an average of $\sigma^2_{X_i-\tilde{X}_i}/\sigma^2_{X_i}$ over the all vector elements as given in Eq. (8).

Note that, when data have autocorrelation, the bias correction dataset can be chosen such that they have smaller autocorrelation than the original data by taking a wide separation in the trajectory direction of the autocorrelation. It makes the bias correction more efficient, suppresses the statistical error increase, and yields $\alpha < 1$.

**Boosting.** In practice, the binary optimization in Eq. (2) is difficult to solve for a large $N_q$. Although a quantum annealer is employed to solve the optimization problem, the maximum number of fully-connected qubits is limited to $\mathcal{O}(100)$ for the current quantum processors. However, the problem can be decomposed into a linear combination of smaller $N_q$ by applying the idea of Boosting[36,37].

Assume that we have a matrix $\boldsymbol{\phi}_1$ and vectors $\{\boldsymbol{a}_1^{(k)}\}$ of $N_{q1}$ binary elements that approximately reconstruct the input vectors $\mathbf{X}^{(k)} \approx \boldsymbol{\phi}_1 \boldsymbol{a}_1^{(k)}$. We can find another set of solutions of $\boldsymbol{\phi}_2$ and $\{\boldsymbol{a}_2^{(k)}\}$ of $N_{q2}$ binary elements reconstructing the reconstruction error of $\boldsymbol{\phi}_1 \boldsymbol{a}_1^{(k)}$ by taking $\mathbf{X}^{(k)} - \boldsymbol{\phi}_1 \boldsymbol{a}_1^{(k)}$ as new input vectors. By combining the two sets of solutions, we can build a precise reconstruction of $\mathbf{X}^{(k)}$ as

$$\mathbf{X}^{(k)} \approx \boldsymbol{\phi}_1 \boldsymbol{a}_1^{(k)} + \boldsymbol{\phi}_2 \boldsymbol{a}_2^{(k)} = \begin{pmatrix} \boldsymbol{\phi}_1 & 0 \\ 0 & \boldsymbol{\phi}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{a}_1^{(k)} \\ \boldsymbol{a}_2^{(k)} \end{pmatrix}, \tag{13}$$

where the total number of binary coefficients representing an input vector $\mathbf{X}^{(k)}$ is $N_{q1} + N_{q2}$. This procedure can be repeated to an arbitrary number of sets of solutions. For an ideal binary optimizer, decomposing the problem into a smaller number of binary elements makes the solution worse than the full solution because the decomposition ignores the correlation between the different sets, that potentially reduces the reconstruction error. For realistic binary optimizers, however, boosting can provide a better solution.

## Numerical experiments

**Test data.** In this study, we use the Monte Carlo simulation data of lattice QCD, a theory of quarks and gluons, and their interactions. The lattice QCD simulations produce large amounts of data that need to be stored for analysis, but the data are correlated with each other, so a data compression algorithm exploiting the correlation can obtain a better compression ability. Among various lattice QCD observables, in this study, we use the three-point correlation function data of nucleon *vector* and *axial-vector (axial)* charges, which describe the response of a nucleon to particles such as the neutrino. For most of the test cases, we shape the data into 10 independent sets of 3200 vectors with 16 components, so each dataset has $N = 3200$ and $D = 16$. As illustrated in Fig. 1, there are strong correlations between the 16 components. Since the *vector* data show a stronger correlation than the *axial-vector* data, we expect the proposed algorithm to give a better compression (smaller $Q^2$) for the *vector* data than the *axial-vector* data. We standardize the data as a pre-processing step to obtain a homogeneous reconstruction error on all 16 components.

**Experiments with simulated annealing.** First, we carry out the demonstration of the proposed compression algorithm with simulated annealing[38,39] on classical computers by employing the D-Wave's simulated annealing sampler implemented in the D-Wave's Ocean library[40]. In the simulated annealing, we take the minimum energy solution from the 150 runs (*num_reads*=150), while all other parameters are set to their default

| $N_q$ | Vector | Axial | | |
|---|---|---|---|---|
| $(N_{q1} + N_{q2})$ | $Q^2$(BC) | $Q^2$(BC) | | |
| 8 | 0.1115(65) | 0.251(11) | | |
| 16 | 0.0156(14) | 0.1062(74) | | |
| 8+8 | 0.0276(42) | 0.1247(64) | | |
| 24 | 0.00404(68) | 0.0620(73) | | |
| 32 | 0.00152(15) | 0.0365(61) | | |
| 16+16 | 0.00164(33) | 0.0396(71) | | |
| 48 | 0.00081(9) | 0.0154(11) | | |
| 24+24 | 0.00016(2) | 0.0151(26) | | |
| 64 | 0.00063(6) | 0.0163(22) | | |
| 32+32 | 0.000052(2) | 0.0047(5) | | |
| | Vector | | Axial | |
| $N_z$ | $Q^2$(PCA) | $Q^2$(AE) | $Q^2$(PCA) | $Q^2$(AE) |
| 1 | 0.326(24) | 0.160(16) | 0.501(13) | 0.250(11) |
| 2 | 0.1103(91) | 0.0243(23) | 0.2760(97) | 0.0822(58) |
| 3 | 0.0356(31) | 0.00454(36) | 0.1803(72) | 0.0419(25) |
| 4 | 0.00021(2) | 0.00019(1) | 0.1073(52) | 0.0256(11) |

**Table 1.** $Q^2$, defined in Eq. (8), of the binary compression (BC) algorithm we propose with $N_q$ qubits (left) and classical approaches of the principal component analysis (PCA) and autoencoder (AE) with $N_z$ codes (right) for the vector and axial-vector data. The results with $N_q = N_{q1} + N_{q2}$ shows the compression with the boosting explained in "Boosting" section. Results are averaged over 10 independent sets, and the errors are calculated as the standard deviation of the mean. A smaller $Q^2$ indicates a better reconstruction.

values. As described in supplementary section 1, we find that the simulated annealing with *num_reads=150* gives close to the exact solution up to around $N_q = 20$, and the quality of the solution deteriorates as $N_q$ is increased.

To find the solutions $\{\boldsymbol{a}^{(k)}\}$ and $\boldsymbol{\phi}$ of the optimization problem in Eq. (2), we iterate the optimization in $\boldsymbol{\phi}$ and $\{\boldsymbol{a}^{(k)}\}$ as described in "Data compression" section. In this study, $\boldsymbol{\phi}$ is updated as follows. After obtaining the solution $\tilde{\boldsymbol{\phi}}$ that minimizes the reconstruction error of the mini-batch using the L-BFGS-B algorithm[41], we update $\boldsymbol{\phi}$ as

$$\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \eta(\tilde{\boldsymbol{\phi}} - \boldsymbol{\phi}). \tag{14}$$

Here the learning rate $\eta$ is continuously decreased from the initial value $\eta_0$ as the number of training epochs ($n_{\text{epoch}}$) is increased, following $\eta = \eta_0 \times 0.8^{n_{\text{epoch}}}$. For the batch size and initial learning rate, we use $N_b = 50$ and $\eta_0 = 0.9$ as we find that those give the best or close to the best results after exploring a grid of $N_b$ and $\eta_0$. The final results are obtained with 30 epochs of training steps.

To compare with the compression performance of the proposed algorithm, we study the conventional data compression algorithms using principal component analysis (PCA) and neural-network-based autoencoder. PCA finds orthogonal directions that maximize the variance as principal components. By saving only the coefficients of the first few principal components, the PCA works as a lossy data compression algorithm. We reconstructed the data from the first $N_z$ principal components to obtain the data compression. Autoencoder also provides data compression by constraining the number of codes ($N_z$) to a small number[42,43]. We used a fully connected neural-network encoder and decoder with three hidden layers of $(D, 128, 64, 32, N_z)$ and $(N_z, 32, 64, 128, D)$ with rectified linear unit (ReLU) activation functions. For the training, we use the Adam optimizer[44] implemented in the PyTorch python library[45] with the learning rate of 0.01 and the batch size of 3200, which are the optimal hyperparameters determined from a grid search. After 5000 epochs of training, we continue the training until we reach a better reconstruction error than the best reconstruction error we have obtained in the first 5000 epochs and stop the training.

The results are summarized in Table 1 and Fig. 2. The results show that the boosting approach gives better results than the full calculation for $N_q > 32$, where the simulated annealing fails in finding the close-to-ground solution. The comparison between different algorithms shows that the autoencoder outperforms the PCA, and the proposed binary compression outperforms the autoencoder. The compression quality ($Q^2$) of the autoencoder with $N_z$ number of codes can be obtained using the proposed binary compression algorithm with the number of bits around $N_q \approx 10N_z$. Considering single-precision floating-point numbers, which usually occupying 32 bits for a number, the proposed algorithm provides the same quality of compression as the autoencoder approach using about 3–3.5 times smaller memory space.

We also carry out a scaling test by observing the data compression performance for different dimensions of the lattice QCD simulation data, $D = 8, 16, 32$, and 48. In this test, the data correlation pattern is kept the same as the one described in Fig. 1: $D$ can be decomposed into four blocks, and the data points within a block have a stronger correlation than the data points between different blocks. Results are presented in Fig. 3. The two upper plots show that the compression algorithm becomes more efficient, which means the smaller reconstruction error for a fixed compression rate, as the data dimension becomes larger. This is because larger dimensional data has

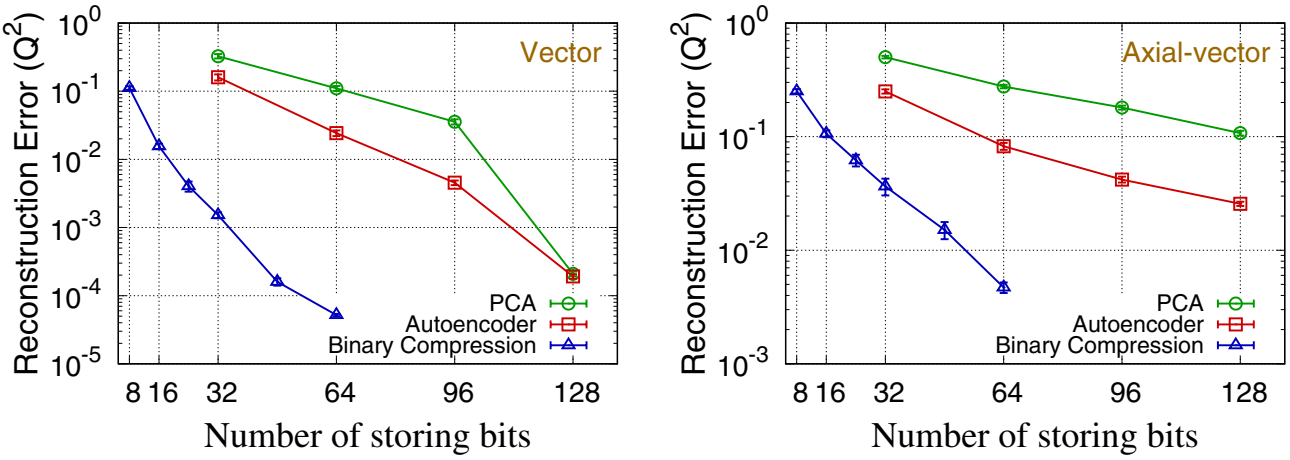



**Figure 2.** $Q^2$, defined in Eq. (8) for different number of storing bits for data dimension $D = 16$. For the principal component analysis (PCA) and autoencoder (AE) approaches, the number of storing bits is calculated by $32 \times N_z$, assuming single-precision floating-point numbers. For binary compression algorithm of $N_q >= 48$, we use the boosting approach with $N_{q1} = N_{q2} = N_q/2$.

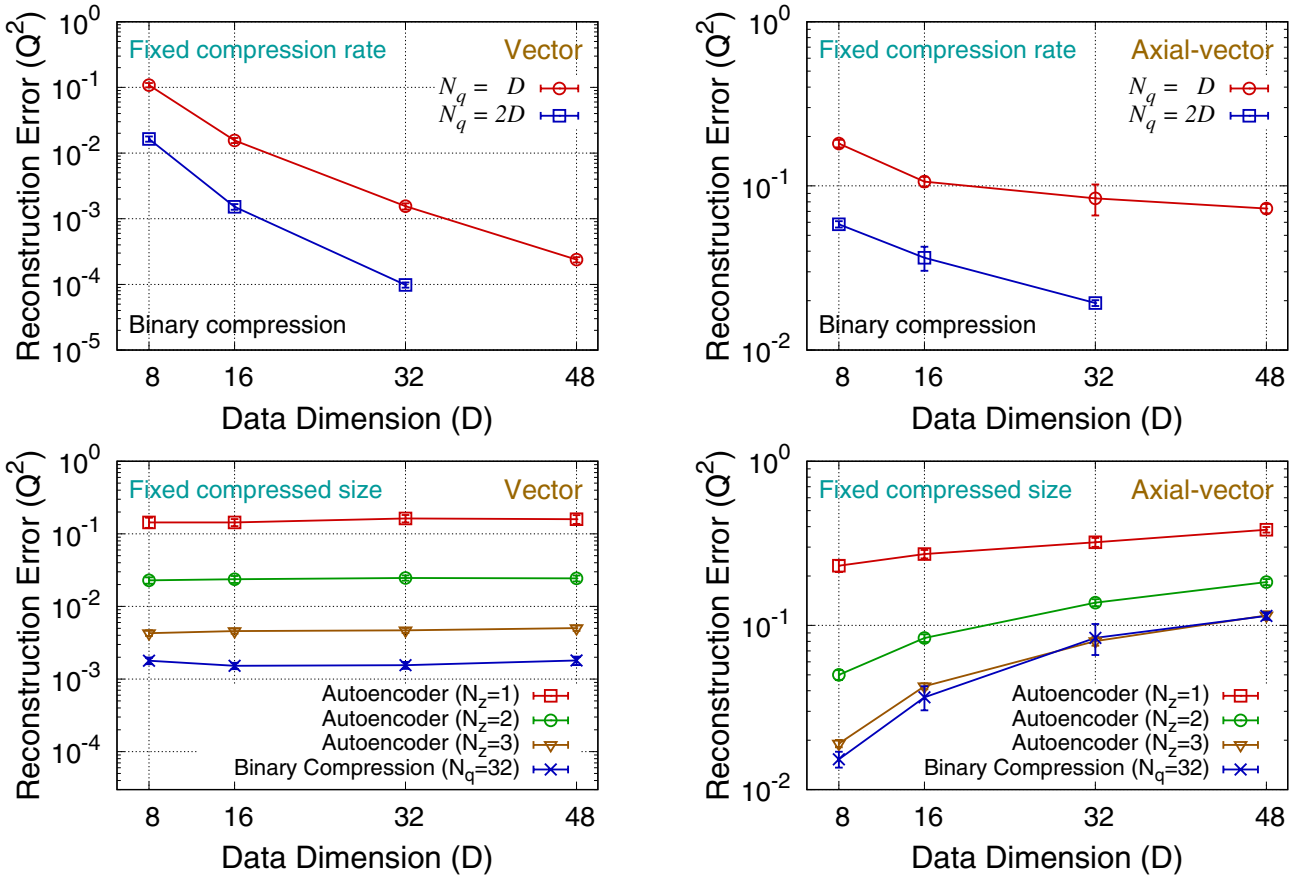



**Figure 3.** Scaling of $Q^2$ for various dimensions ($D$) vector (left) and axial-vector (right) data. Upper figures show the results using the binary compression algorithm at fixed compression rates ($N_q = D$ and $N_q = 2D$), and the bottom figures show the comparison between the binary compression and autoencoder algorithms at fixed sizes in the compressed space ($N_z = 1, 2, 3$ for autoencoder, and $N_q = 32$ for binary compression). For binary compression algorithm of $N_q >= 48$, we use the boosting approach with $N_{q1} = N_{q2} = N_q/2$.

| | $Q^2$ (Vector) | | | |
|---|---|---|---|---|
| | Free $\phi$ | | $\max(\lvert\phi_{ij}\rvert) = 1$ | |
| $N_q$ | D-Wave | Sim.Ann. | D-Wave | Sim.Ann. |
| 8 | 0.104(11) | 0.104(11) | 0.099(10) | 0.099(10) |
| 16 | 0.0124(16) | 0.0120(16) | 0.0192(25) | 0.0197(27) |
| 32 | 0.0068(12) | 0.0014(02) | 0.0046(10) | 0.0033(06) |
| 48 | 0.0066(10) | 0.0007(01) | 0.0048(11) | 0.0015(03) |
| 60 | 0.0099(19) | 0.0007(01) | 0.0025(04) | 0.0006(01) |
| | $Q^2$ (Axial) | | | |
| | Free $\phi$ | | $\max(\lvert\phi_{ij}\rvert) = 1$ | |
| $N_q$ | D-Wave | Sim.Ann. | D-Wave | Sim.Ann. |
| 8 | 0.289(22) | 0.289(22) | 0.297(30) | 0.297(30) |
| 16 | 0.1117(87) | 0.1101(87) | 0.129(20) | 0.135(20) |
| 32 | 0.1113(86) | 0.0366(50) | 0.090(11) | 0.072(12) |
| 48 | 0.092(15) | 0.0214(34) | 0.0751(71) | 0.0303(39) |
| 60 | 0.0962(70) | 0.0175(16) | 0.0886(86) | 0.0221(20) |

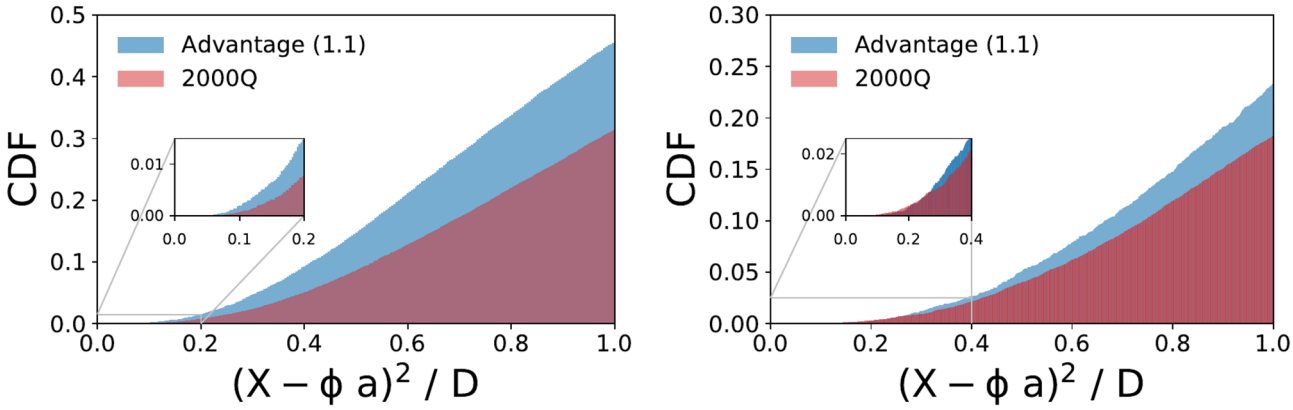
**Table 2.** $Q^2$ values of the binary compression algorithm on D-Wave 2000Q quantum annealer (D-Wave) and the simulated annealing (Sim.Ann.) with ($\max(|\boldsymbol{\phi}_{ij}|) = 1$) and without (Free $\boldsymbol{\phi}$) the constraints on the elements of $\boldsymbol{\phi}$. Results are obtained from a set of $N = 200$ vector and axial-vector data. Numbers in the parenthesis are the statistical error of the 200 samples estimated by the bootstrap method[47].

a larger number of correlated data components that can be exploited for efficient data compression. The two bottom plots show that the data compression performance of the binary compression algorithm with $N_q = 32$ is similar to that of the autoencoder with $N_z = 3{-}3.5$ for all different data dimensions. These results support that, considering single-precision floating-point numbers for the autoencoder codes, the binary compression is about 3–3.5 times more efficient than the autoencoder. Note that, however, this performance gain should depend on the correlation pattern of the data.

**Experiments with D-Wave 2000Q.** To verify the usability of the existing quantum hardware for the proposed compression algorithm, we carry out the $\boldsymbol{a}^{(k)}$-optimization of Eq. (2) using the D-Wave 2000Q quantum processor with the $\boldsymbol{\phi}$ obtained using the simulated annealing as described in "Experiments with simulated annealing" section. The major issue with the D-Wave quantum annealer is that the $h$ and $J$ parameters have poor precision when implemented in the D-Wave QPU, even though they are specified as double-precision floating-point numbers in the program, due to the integrated control errors[46]. Since the parameters are normalized by the maximum absolute value of $h$ and $J$, small-value elements are more sensitive to the integrated control errors. When there are large scale variations in the $h$ and $J$ parameters, furthermore, the information of the small-value elements could be washed off due to the errors, and the annealing results may be different from the true solution of the original problem.

In the data compression problem, to minimize the effect of the fidelity loss in the final results, we restrict the maximum absolute value of the matrix elements of $\boldsymbol{\phi}$ by 1. It prevents a large maximum absolute value of $h$ and $J$, which introduces a large distortion of the small-value elements. Due to the limited D-Wave access time, we carry out the study only for one set of $N = 200$ samples. For this study, we obtained solutions to this QUBO problem on the D-Wave 2000Q quantum hardware that are drawn from 5000 reads using a series of 20 different chain strengths within the range (2.0, 3.0). Results are taken from the lowest energy points among overall $5000 \times 20$ solutions of the 2000Q machine. The embedding procedure is repeated only for a new input but was kept unchanged as we changed the chain strength values. In a control run, we find that even if one runs a new embedding each time a new chain strength changes, the final results do not differ from the method described above. However, the later approach that requires a new embedding solution for each chain strength will require more pre-processing time.

Table 2 shows the $Q^2$ values of the binary compression algorithm on D-Wave 2000Q in comparison with the simulated annealing optimizer. When $N_q \leq 16$, D-Wave shows similar performance as the simulated annealing, but when $N_q > 16$, D-Wave shows worse performance than the simulated annealing. The reconstruction error, represented by $Q^2$, is decreased as $N_q$ is increased on the simulated annealing, but no significant decrease of the $Q^2$ is observed on the D-Wave for $N_q > 32$ compared to the results from $N_q = 32$. As expected, constraining $\max(|\boldsymbol{\phi}_{ij}|) = 1$ improves the results on the D-Wave for $N_q \geq 32$, but the D-Wave results are still worse than the simulated annealing. Note that, due to the limited D-Wave access time, the results were obtained with a fixed $\boldsymbol{\phi}$ obtained using the simulated annealing. Hence, the results show a comparison of the optimization performance for a given problem. If $\boldsymbol{\phi}$ were obtained directly from the D-Wave quantum annealer, however, optimal constraints to meet the hardware limitations would have been imposed, naturally, and it might have resulted in a better compression performance than those of the $\max(|\boldsymbol{\phi}_{ij}|) = 1$ constraints. Note that the quality of the solutions from the D-Wave quantum annealer is dependent on the nature of the data. In general, we expect better D-Wave quantum annealing results for a dataset that gives smaller scale variations in the $h$ and $J$ parameters through Eq. (4).




(a) $N_q = 32$ (b) $N_q = 60$

**Figure 4.** Cumulative distribution function (CDF) of the normalized reconstruction error from all feasible samples obtained from the D-Wave 2000Q (red) and Advantage system (blue) for the axial-vector data. About 50% and 38% of the samples were feasible from D-Wave 2000Q and Advantage for $N_q = 32$, respectively. For $N_q = 60$ there were about 51% and 18%, respectively.

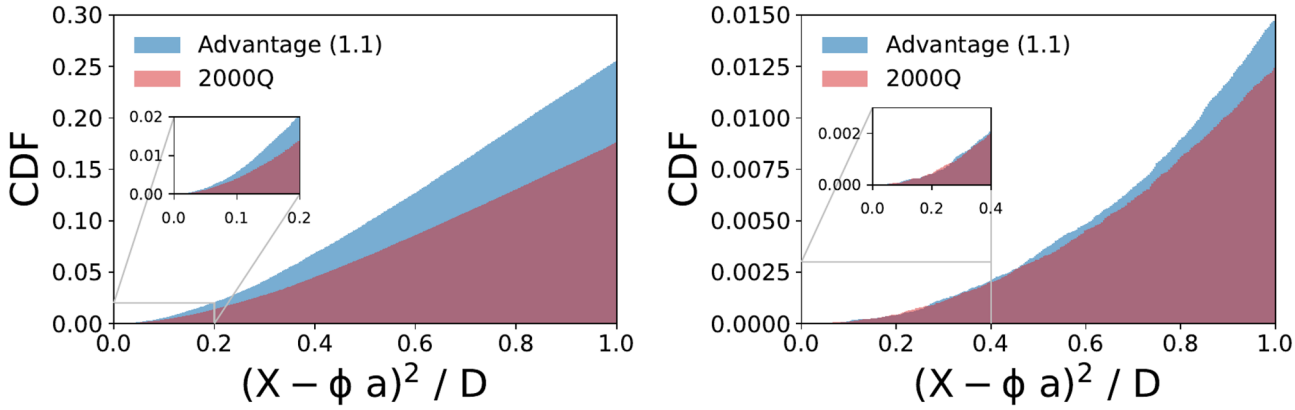
**Comparison of D-Wave 2000Q with advantage systems.** We benchmark the D-Wave Advantage system in comparison with the 2000Q using the $\boldsymbol{a}^{(k)}$-optimization problem in Eq. (2). For axial and vector data we compute the cumulative distribution function (CDF) of the normalized reconstruction error for systems of size $N_q = (32, 60)$. To minimize possible biases due to a specific choice of embedding, we employ the heuristic solvers provided by Dwave to find an embedding for each configuration and proceed to collect at least 1500 samples (per configuration). The chain strength during embedding was determined by the maximal coupling in absolute value, multiplied by a hyperparameter which we call chain strength multiple. The number of physical qubits, in practice, is many times higher than the logical qubits required, due to hardware connectivity. There are cases where the physical qubits that are strongly coupled to behave as one logical qubit, return different values and we discard these samples, as non viable solutions, from our calculation of the distribution function. (For axial data, $N_q = 32$: 150 qubits Advantage/350 qubits 2000Q, $N_q = 60$: 600 qubits Advantage/1600 qubits 2000Q. For vector data, $N_q = 32$: around 180 qubits Advantage/380 qubits 2000Q, $N_q = 60$: 600 qubits Advantage/1400 qubits 2000Q). As the number of qubits increases the fraction of feasible samples decreases. Also, the fraction of the CDF with small reconstruction error decreases. By trial and error, we find that setting chain strength multiple to a value greater than 1 reduces the number of viable solutions from the Advantage system, but it improves the results from 2000Q. For the samples collected for axial data, we set them to 0.8 and 1.6 respectively.

As can be seen from Figs. 4 and 5, when $N_q = 32$, both hardware perform rather well, and the new Advantage system has better statistics and overall higher quality of sub optimal solutions. In the case of $N_q = 60$, the difference between the two hardware becomes less distinct and the CDF is peaked on solutions with high reconstruction error. As we did not apply boosting for these experiments, the quality degradation as the number of qubits increased can be ascribed to the connectivity of the hardware.

## Discussion

In this paper, we presented a new lossy compression algorithm for statistical data based on the representation learning with binary variables. The algorithm finds a set of basis vectors, which is common for all data, and their binary coefficients ($N_q$) that precisely reconstruct each $D$-dimensional input vector. The algorithm provides data compression because the $N_q$-dimensional binary representation requires much smaller storage space than the original data of $D$-dimensional floating-point numbers. We also presented a bias correction procedure estimating the errors due to the inexact reconstruction of the lossy compression in "Bias correction" section. The compression algorithm was applied to two lattice QCD datasets in "Numerical experiments" section. With simulated annealing, the binary compression algorithm was able to achieve the same quality of reconstruction with 3–3.5 times smaller storage usage than the algorithm using neural-network autoencoder. The binary optimization carried out on D-Wave 2000Q for the compression problems showed promising results, but the performance was limited by the integrated control error of the D-Wave QPU, which introduces large uncertainties in the $h$ and $J$ parameters. The comparison of D-Wave 2000Q and Advantage systems showed that the Advantage is more efficient than the 2000Q in obtaining the low-energy solutions.

The proposed compression algorithm is a natural outlier detector because input data with large reconstruction errors can be marked anomalous[48]. Using the proposed algorithm, furthermore, many operations that need to be performed on the floating point numbers $\mathbf{X}^{(k)}$ can be replaced by those on single-bit coefficients $\boldsymbol{a}^{(k)}$ with much smaller computational cost, because the relationship between $\mathbf{X}^{(k)}$ and $\boldsymbol{a}^{(k)}$ is linear ($\mathbf{X}^{(k)} \approx \boldsymbol{\phi}\boldsymbol{a}^{(k)}$), and the single-bit coefficients satisfy $\left(a_j^{(k)}\right)^n = a_j^{(k)}$ for any $n$, which simplifies power operations. Here are two examples of the operations in the compressed space:




(a) $N_q = 32$ (b) $N_q = 60$

**Figure 5.** Cumulative distribution function (CDF) of the normalized reconstruction error from all feasible samples obtained from the D-Wave 2000Q (red) and Advantage system (blue) for the vector data. About 95% and 91% of the samples were feasible from D-Wave 2000Q and Advantage for $N_q = 32$, respectively. For $N_q = 60$ there were about 63% and 73%, respectively.

- Sum of vectors

$$\sum_{k=1}^{N} \mathbf{X}^{(k)} \approx \sum_{k=1}^{N} \boldsymbol{\phi} \boldsymbol{a}^{(k)} = \boldsymbol{\phi} \left( \sum_{k=1}^{N} \boldsymbol{a}^{(k)} \right), \tag{15}$$

- Sum of $l^2$-norm squares

$$\begin{aligned} \sum_{k=1}^{N} ||\mathbf{X}^{(k)}||^2 &\approx \sum_{k=1}^{N} \sum_{i=1}^{D} \left( \sum_{j=1}^{N_q} \phi_{ij} a_j^{(k)} \right)^2 \\ &= \sum_{i=1}^{D} \left[ \sum_{j=1}^{N_q} \phi_{ij}^2 \left( \sum_{k=1}^{N} a_j^{(k)} \right) + 2 \sum_{l<m} \left( \sum_{k=1}^{N} a_l^{(k)} a_m^{(k)} \right) \phi_{il} \phi_{im} \right]. \end{aligned} \tag{16}$$

The cost reduction is maximized when $D, N_q \ll N$, which is a typical case of many statistical datasets.

In this study, we presented only the results with the $\boldsymbol{\phi}$ calculated from the whole dataset. In general, however, $\boldsymbol{\phi}$ obtained from a smaller subset of the whole data provides a reasonably good compression performance. When using a $\boldsymbol{\phi}$ obtained from a subset data, some unseen data vectors could yield large reconstruction error. To control the error and maintain the quality of the compression, one needs to define a threshold and save the original data when the data gives a reconstruction error bigger than the threshold.

Received: 16 November 2021; Accepted: 10 February 2022
Published online: 09 March 2022

## References

1. Park, S., Gupta, R., Yoon, B., Mondal, S., Bhattacharya, T., Jang, Y.-C., Joó, B. & F. Winter *Precision Nucleon Charges and Form Factors Using 2+1-flavor Lattice QCD* (Nucleon Matrix Elements (NME), 2021). arXiv:2103.05599 [hep-lat]
2. He, J. *et al.* Detailed analysis of excited state systematics in a lattice QCD calculation of $g_A$ (2021). arXiv:2104.05226 [hep-lat]
3. Lakshminarasimhan, S., Shah, N., Ethier, S., Ku, S.-H., Chang, C. S., Klasky, S., Latham, R., Ross, R. & Samatova, N. F. ISABELA for effective in situ compression of scientific data: Isabela for effective in-situ reduction of spatio-temporal data. *Concurr. Comput. Pract. Exp.* **25** (2012). https://doi.org/10.1002/cpe.2887
4. Lindstrom, P. Fixed-rate compressed floating-point arrays. *IEEE Trans. Vis. Comput. Graph.* **20**, 2674 (2014).
5. Di, S. & Cappello, F. Fast error-bounded lossy HPC data compression with SZ, in *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 730–739 (2016).
6. Tao, D., Di, S., Chen, Z. & Cappello, F. Significantly improving lossy compression for scientific data sets based on multidimensional prediction and error-controlled quantization. in *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)* (2017). https://doi.org/10.1109/ipdps.2017.115
7. Tian, J., Di, S., Zhao, K., Rivera, C., Fulp, M. H., Underwood, R., Jin, S., Liang, X., Calhoun, J., Tao, D. & Cappello, F. CuSZ: An efficient gpu-based error-bounded lossy compression framework for scientific data. in *Proceedings of the ACM International Conference on Parallel Architectures and Compilation Techniques*, PACT '20, 3–15 (Association for Computing Machinery, 2020).
8. Chen, Z., Son, S. W., Hendrix, W., Agrawal, A., Liao, W.-K. & Choudhary, A. NUMARCK: Machine learning algorithm for resiliency and checkpointing. in *SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 733–744 (2014).
9. Chou, J. & Piegl, L. Data reduction using cubic rational B-splines. *IEEE Comput. Graph. Appl.* **12**, 60 (1992).

10. Jin, R., Goswami, A. & Agrawal, G. Fast and exact out-of-core and distributed k-means clustering. *Knowl. Inf. Syst.* **10**, 17 (2006).
11. Glaws, A., King, R. & Sprague, M. Deep learning for in situ data compression of large turbulent flow simulations. *Phys. Rev. Fluids* **5**, 114602 (2020).
12. Wang, S., Chen, H., Wu, L. & Wang, J. A novel smart meter data compression method via stacked convolutional sparse auto-encoder. *Int. J. Electr. Power Energy Syst.* **118**, 105761 (2020).
13. Romero, J., Olson, J. P. & Aspuru-Guzik, A. Quantum autoencoders for efficient compression of quantum data. *Quant. Sci. Technol.* **2**, 045001 (2017).
14. Deutsch, D. Quantum computational networks. *Proc. R. Soc. Lond. Ser. A* **425**, 73 (1989).
15. Barenco, A. *et al.* Elementary gates for quantum computation. *Phys. Rev. A* **52**, 3457–3467 (1995).
16. Farhi, E., Goldstone, J., Gutmann, S. & Sipser, M. *Quantum Computation by Adiabatic Evolution* (2000). arXiv:quant-ph/0001106
17. Das, A. & Chakrabarti, B. K. Colloquium: Quantum annealing and analog quantum computation. *Rev. Mod. Phys.* **80**, 1061 (2008).
18. Albash, T. & Lidar, D. A. Adiabatic quantum computation. *Rev. Mod. Phys.* **90**, 015002 (2018).
19. de Falco, D., Apolloni, B. & Cesa-Bianchi, N. A numerical implementation of quantum annealing, p. 97 (1988)
20. Ray, P., Chakrabarti, B. K. & Chakrabarti, A. Sherrington–Kirkpatrick model in a transverse field: Absence of replica symmetry breaking due to quantum fluctuations. *Phys. Rev. B* **39**, 11828 (1989).
21. Kadowaki, T. & Nishimori, H. Quantum annealing in the transverse Ising model. *Phys. Rev. E* **58**, 5355 (1998).
22. Santoro, G. E., Martoňák, R., Tosatti, E. & Car, R. Theory of quantum annealing of an Ising spin glass. Science **295**, 2427–2430 (2002)
23. Mukherjee, S. & Chakrabarti, B. Multivariable optimization: Quantum annealing and computation. *Eur. Phys. J. Spec. Top.* **224**, 17–24 (2015).
24. Hauke, P., Katzgraber, H. G., Lechner, W., Nishimori, H. & Oliver, W. D. Perspectives of quantum annealing: Methods and implementations. *Rep. Prog. Phys.* **83**, 054401 (2020).
25. Lucas, A. Ising formulations of many NP problems. *Front. Phys.* **2** (2014). https://doi.org/10.3389/fphy.2014.00005
26. Farhi, E. *et al.* A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science* **292**, 472–475 (2001).
27. Perdomo-Ortiz, A., Dickson, N., Drew-Brook, M., Rose, G. & Aspuru-Guzik, A. Finding low-energy conformations of lattice protein models by quantum annealing. *Sci. Rep.* **2**, 571 (2012) arXiv:1204.5485 [quant-ph].
28. Rieffel, E. G. *et al.* A case study in programming a quantum annealer for hard operational planning problems. *Quant. Inf. Process.* **14**, 1–36 (2014).
29. Nguyen, N. T. T., Kenyon, G. T. & Yoon, B. A regression algorithm for accelerated lattice QCD that exploits sparse inference on the D-Wave quantum annealer. *Sci. Rep.* **10**, 10915 (2020) arXiv:1911.06267 [quant-ph].
30. "D-Wave System Documentation". https://docs.dwavesys.com/docs/latest/index.html (a). Accessed 10 May 2021.
31. Nguyen, N. T. T. & Kenyon, G. T. 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*, 43–44 (2016).
32. Nguyen, N. T. T., Larson, A. E. & Kenyon, G. T. Generating sparse representations using quantum annealing: Comparison to classical algorithms. in *2017 IEEE International Conference on Rebooting Computing (ICRC)*, 1–6 (2017)
33. Nguyen, N. T. T. & Kenyon, G. T. Image classification using quantum inference on the D-wave 2X. in *2018 IEEE International Conference on Rebooting Computing (ICRC)*, 1–7 (2018). arXiv:1905.13215
34. Yoon, B., Bhattacharya, T. & Gupta, R. Machine learning estimators for lattice QCD observables. *Phys. Rev. D* **100**, 014504 (2019) arXiv:1807.05971 [hep-lat].
35. Zhang, R., Fan, Z., Li, R., Lin, H.-W. & Yoon, B. Machine-learning prediction for quasiparton distribution function matrix elements. *Phys. Rev. D* **101**, 034516 (2020) arXiv:1909.10990 [hep-lat].
36. Kearns, M. Thoughts on hypothesis boosting (1988) **(unpublished)**.
37. Freund, Y. & Schapire, R. E. A short introduction to boosting. in *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1401–1406 (Morgan Kaufmann, 1999).
38. Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Sci. New Ser.* **220**, 671 (1983).
39. Geman, S. & Geman, D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 721 (1984).
40. "D-Wave Ocean Software". https://docs.ocean.dwavesys.com/ (b). Accessed 10 May 2021
41. Zhu, C., Byrd, R. H., Lu, P. & Nocedal, J. *L-BFGS-B—Fortran Subroutines for Large-Scale Bound Constrained Optimization*, Tech. Rep. (ACM Trans. Math. Software, 1994).
42. Theis, L., Shi, W., Cunningham, A. & Huszár, F. Lossy image compression with compressive autoencoders (2017). arXiv:1703.00395 [stat.ML]
43. Watkins, Y., Iaroshenko, O., Sayeh, M. & Kenyon, G. Image compression: Sparse coding vs. bottleneck autoencoders. in *2018 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, 17–20 (2018).
44. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization (2017). arXiv:1412.6980 [cs.LG]
45. ...Paszke, A. *et al.* PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems* Vol. 32 (eds Wallach, H. *et al.*) 8024–8035 (Curran Associates Inc., 2019).
46. "ICE: Dynamic Ranges in h and J Values". https://docs.dwavesys.com/docs/latest/c_qpu_1.html (c). Accessed 10 May 2021.
47. Efron, B. Bootstrap methods: Another look at the jackknife. In *Breakthroughs in Statistics: Methodology and Distribution* (eds Kotz, S. & Johnson, N. L.) 569–593 (Springer, 1992).
48. Adler, A., Elad, M., Hel-Or, Y. & Rivlin, E. Sparse coding with anomaly detection. *J. Signal Process. Syst.* **79**, 179 (2015).


## Acknowledgements

The quantum annealing 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, and the simulated annealing QUBO optimizations were carried out using LANL Institutional Computing cluster and LANL Darwin cluster. Simulation data used for the numerical experiment were generated using the computer facilities at (i)the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231; and, (ii) the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725; (iii) the USQCD Collaboration, which is funded by the Office of Science of the U.S. Department of Energy, (iv) Institutional Computing at Los Alamos National Laboratory. This work was supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics under Contract No. 89233218CNA000001 and LANL Laboratory Directed Research and Development (LDRD) program. Lawrence Berkeley National Laboratory (LBNL) is operated by The Regents of the University of California (UC) for the U.S. Department of Energy (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).

## Author contributions

B.Y. proposed the initial idea and carried out the numerical experiments with simulated annealing. N.N. conducted the numerical experiments on LANL D-Wave 2000Q, and C.C. and E.R. conducted the numerical experiments for the comparison of D-Wave 2000Q and Advantage Systems. All authors contributed to interpreting the results and writing the manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary Information** The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-07539-z.

**Correspondence** and requests for materials should be addressed to B.Y.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2022