Los Alamos NATIONAL LABORATORY

LA-UR-21-29432

Machine Learning Prediction and Compression of Lattice QCD Observables

Boram Yoon

(Los Alamos National Laboratory)

ECT* workshop: Machine Learning for High Energy Physics, on and off the Lattice, Sep 27 – Oct 1, 2021, ECT*, Italy

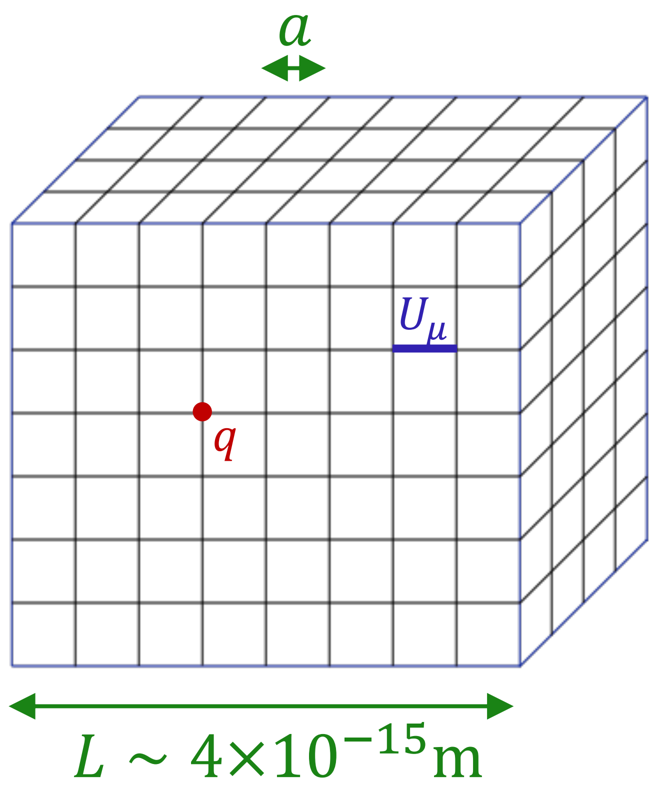
Contents

1) Correlations in Lattice QCD Observables
2) Prediction of Lattice QCD Observables using ML
3) ML Regression Algorithm using D-Wave Quantum Annelaer
4) Lossy Data Compression Algorithm for Lattice QCD Data
5) Summary

Correlations in Lattice QCD Observables

Lattice QCD

- Non-perturbative approach to solving QCD on discretized Euclidean space-time
  - Hypercubic lattice
  - Lattice spacing a
  - Quark fields placed on sites
  - Gluon fields on the links between sites; U_μ
- Numerical lattice QCD calculations using Monte Carlo methods
  - Computationally intensive
  - Use supercomputers
- Continuum results are obtained in $a \to 0$
- Has been successful for many QCD observables
  - Some results are with less than 1% error



Lattice QCD

- Correlation functions

$$\langle O \rangle = Z^{-1} \int dU dq d\bar{q} \, O(U, q, \bar{q}) e^{-S_g - \bar{q}(D+m_q)q}$$

$$= Z^{-1} \int dU \left[O\left(U, (D + m_q)^{-1}\right) e^{-S_g} \det(D + m_q) \right]$$

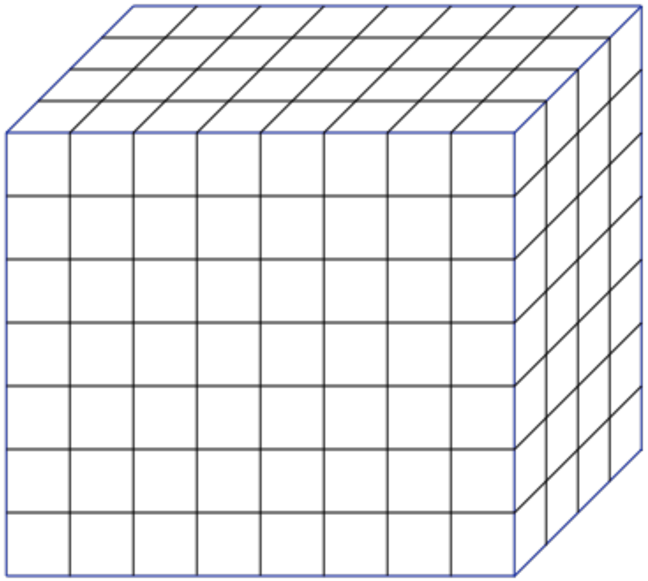
- Monte-Carlo integration
  - Integration variable U is huge

$$N_s^3 \times N_t \times 4 \times 8 \sim 10^9$$

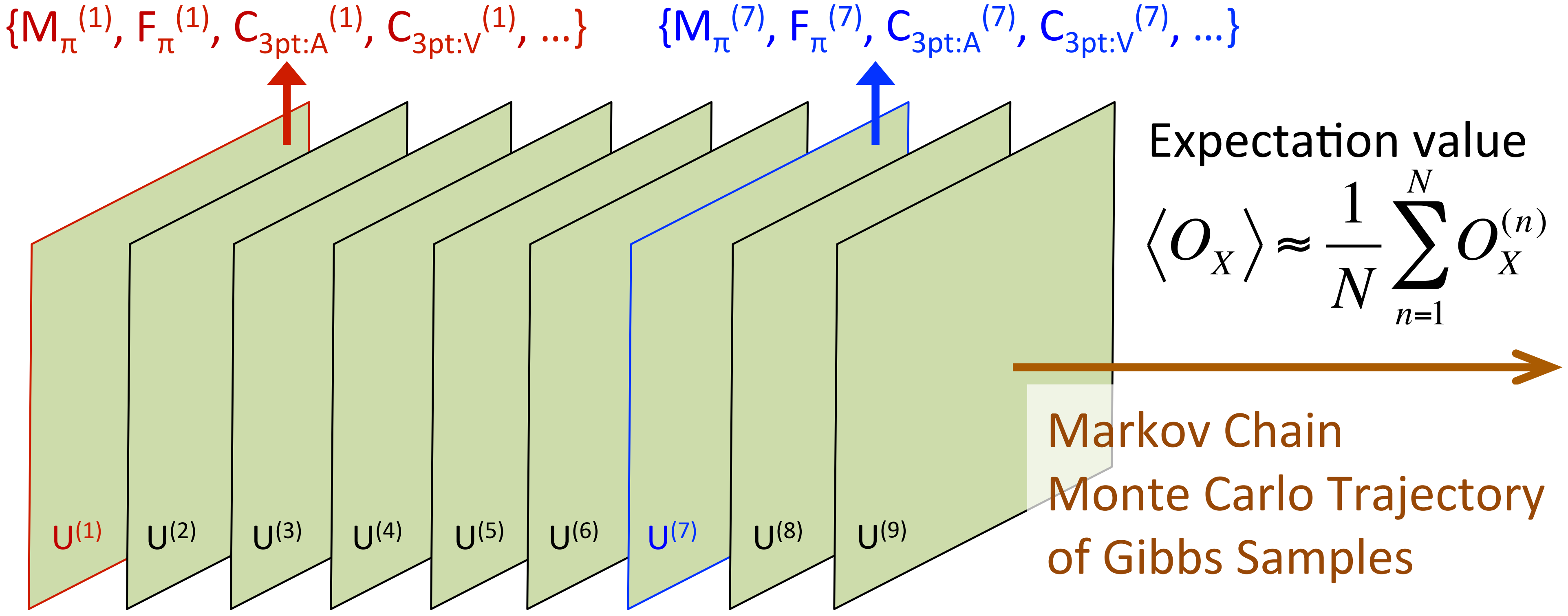
  - Generate Markov chain of gauge configurations U
  - Calculate average as expectation value

$$\langle O \rangle \approx \frac{1}{N} \sum_i O_i \left(U, (D + m_q)^{-1}\right)$$

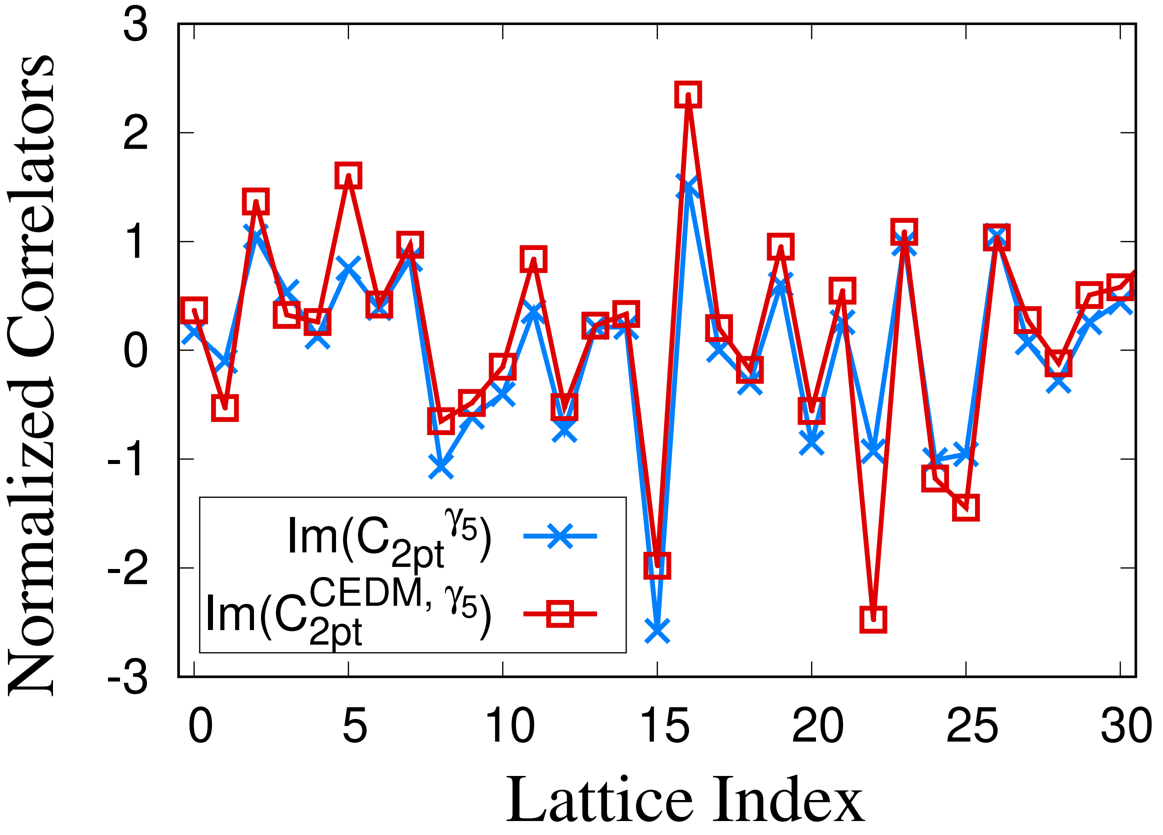
  - Calculation of $O_i\left(U, (D + m_q)^{-1}\right)$: measurement
  - $(D + m)^{-1}$ is computationally expensive



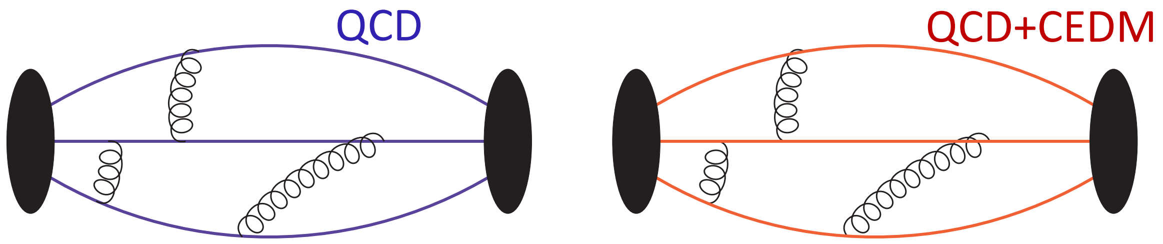
Lattice QCD Observables are Correlated



Correlation Map of Nucleon Observables



- Correlation between proton(uud) 2-pt correlation function and that calculated in presence of CEDM interaction

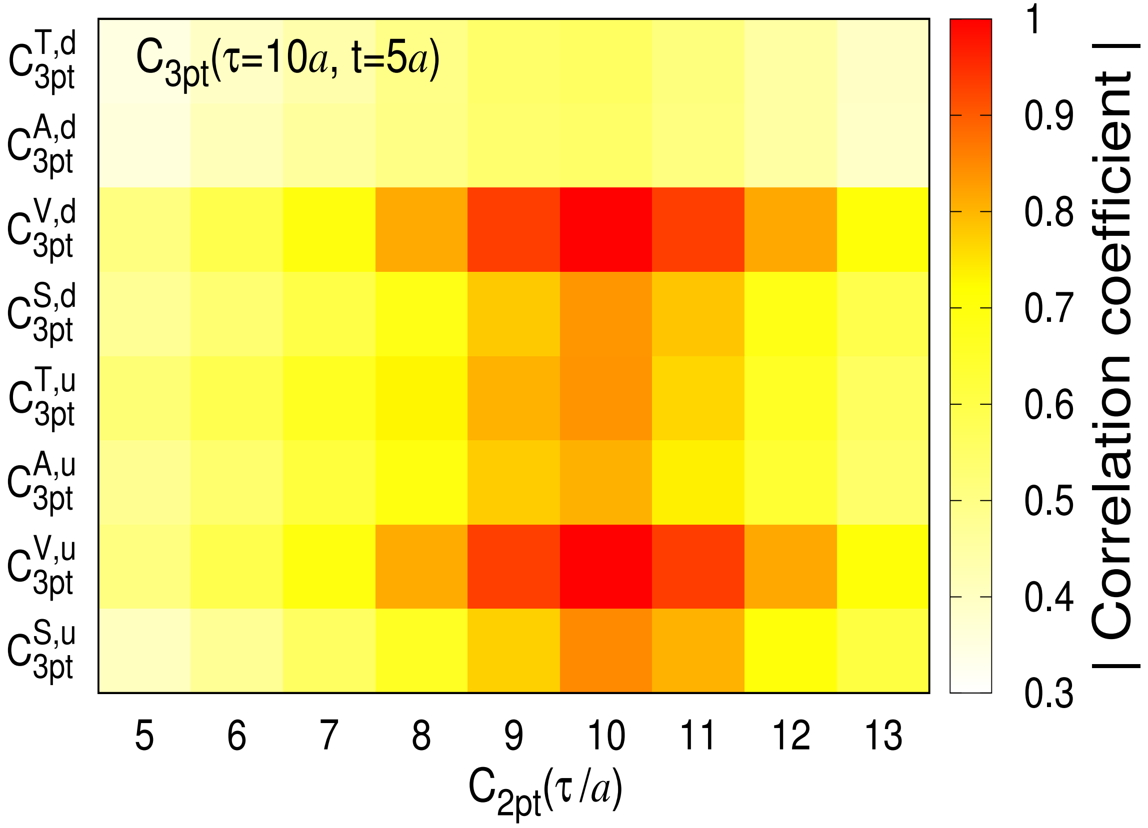


$C_{2pt} \sim \langle N(\tau)N^{\dagger}(0)\rangle$ $\quad C_{2pt}^{CEDM} \sim \langle N(\tau)N^{\dagger}(0)\rangle_{CEDM}$

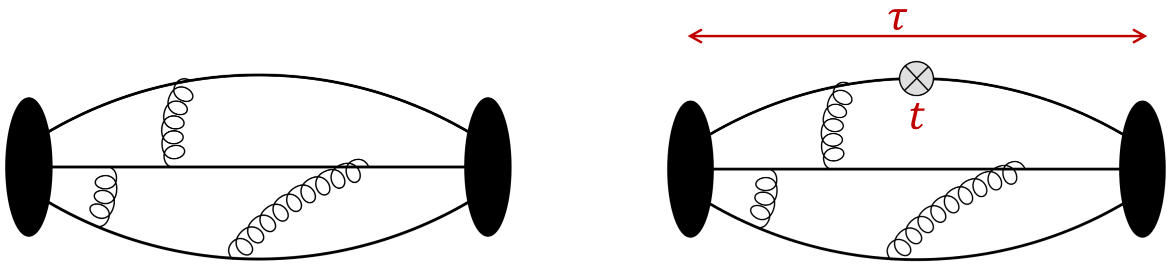
- QCD: D_{clov}

  QCD+CEDM: $D_{clov} + \frac{i}{2}\varepsilon\sigma^{\mu\nu}\gamma_5 G_{\mu\nu}$

Correlation Map of Nucleon Observables



- Correlation between proton(uud) 3-pt and 2-pt correlation functions



$C_{2pt} \sim \langle N(\tau) N^{\dagger}(0) \rangle$ $\quad$ $C_{3pt}^{A,S,T,V} \sim \langle N(\tau) O(t) N^{\dagger}(0) \rangle$

- Using these correlations, C_{3pt} can be estimated from C_{2pt} on each configuration

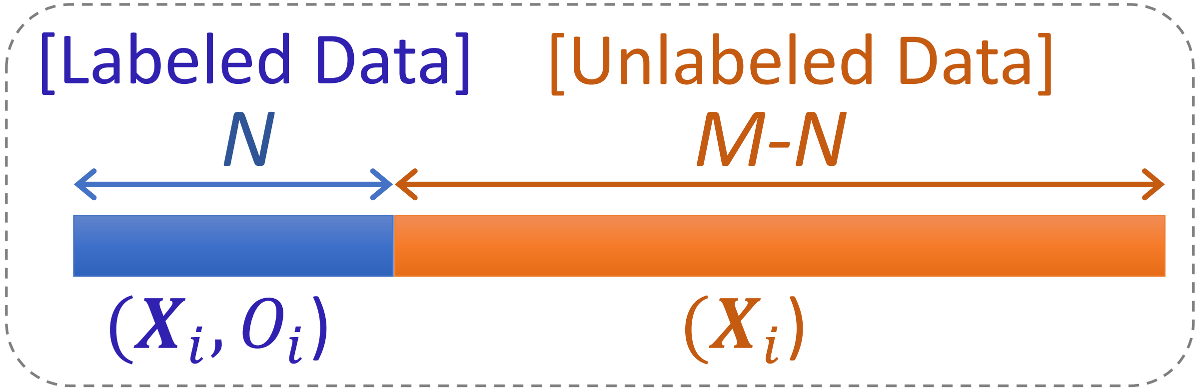
Prediction of Lattice QCD Observables using ML

Measured and computationally cheap observables

→ Prediction of unmeasured and computationally expensive observables

Prediction of Lattice QCD Observables using ML

- Assume *M* indep. measurements
- Common observables $\boldsymbol{X}_i$ on all *M*
  Target observable O_i on first *N*

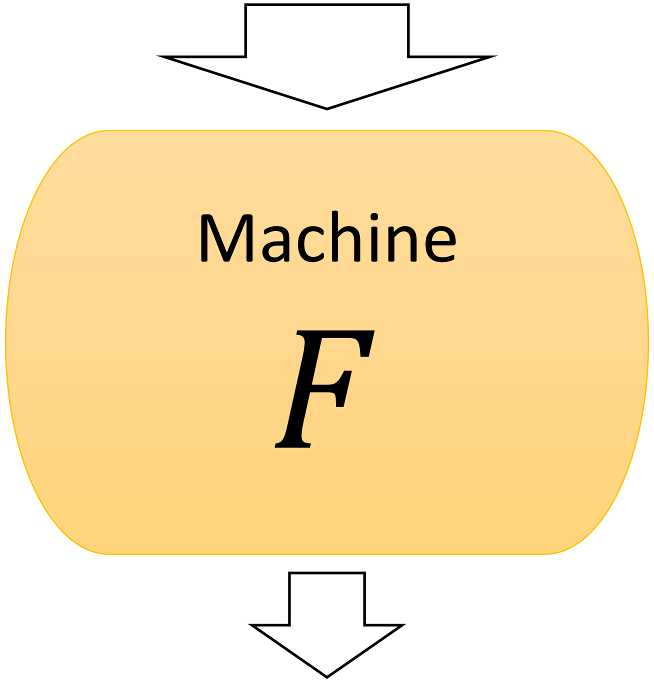


1) **Train** machine ***F*** to yield O_i from $\boldsymbol{X}_i$ on the Labeled Data

2) **Predict** O_i of the Unlabeled data from $\boldsymbol{X}_i$

$$\boldsymbol{F}(\boldsymbol{X}_i) = O_i^P \approx O_i$$

Input: $\boldsymbol{X}_i = (o_i^1, o_i^2, o_i^3, \dots)$



Output: O_i

Prediction Bias

- $\boldsymbol{F}(\boldsymbol{X}_i) = O_i^P \approx O_i$
- Simple average

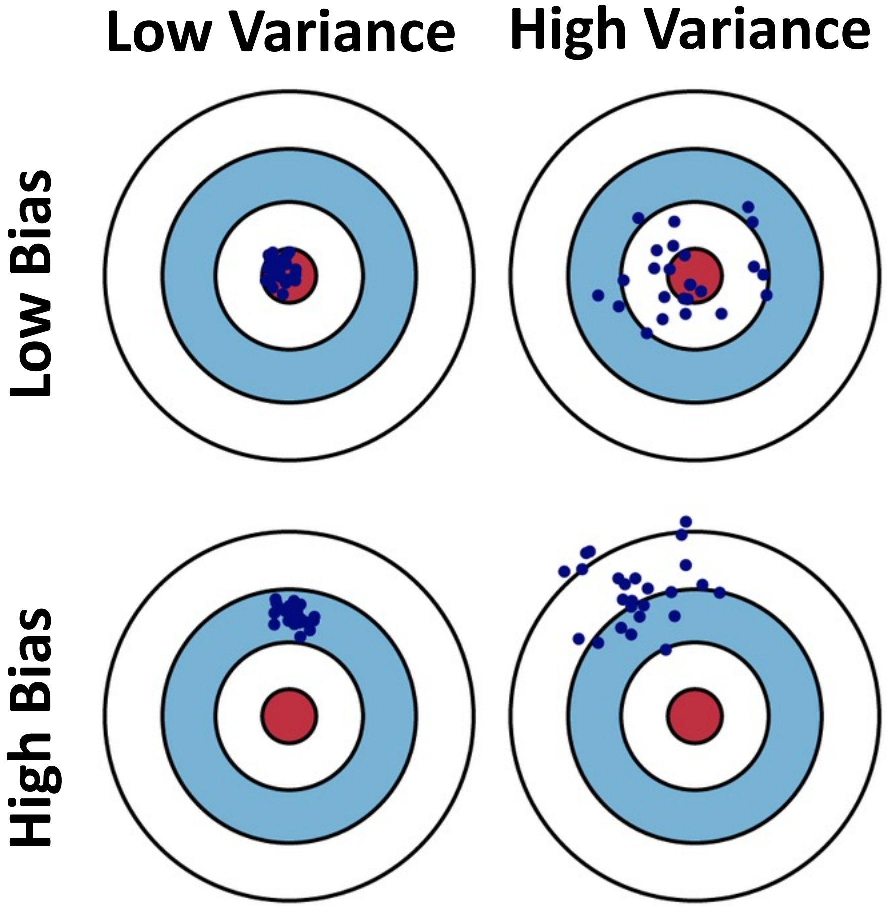
$$\bar{O} = \frac{1}{M} \sum_{i \in \text{Unlabeled}} O_i^P$$

is not correct due to prediction bias

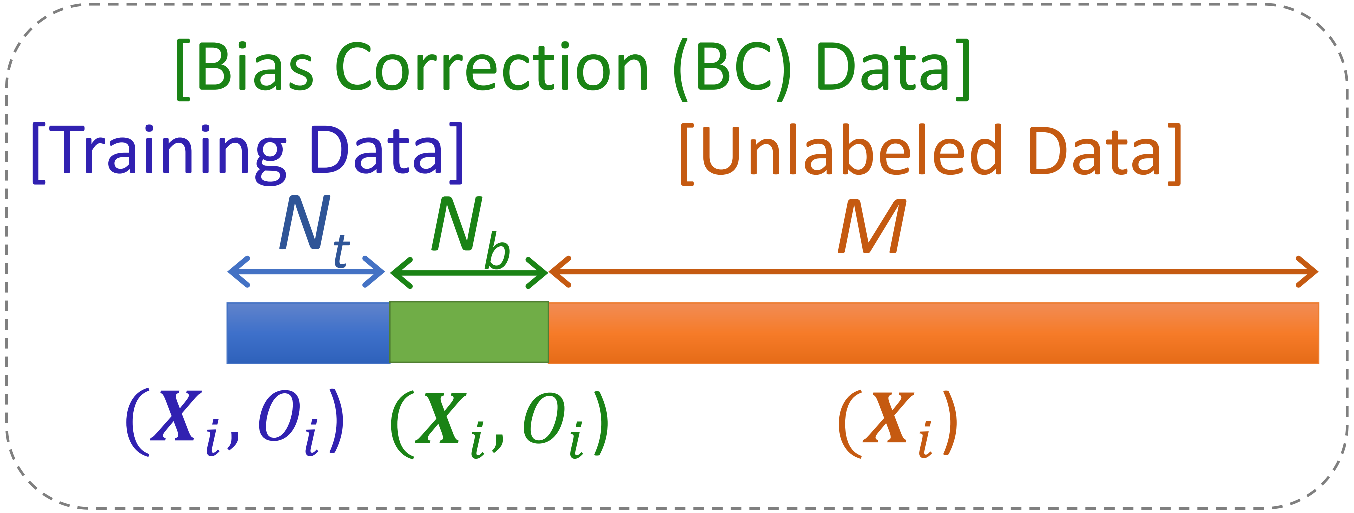
- Prediction = TrueAnswer + Noise + Bias
- ML prediction may have bias

$$\langle O^P \rangle \neq \langle O \rangle$$

$$\text{Bias} = \langle O^P \rangle - \langle O \rangle$$



Bias Correction and Error Quantification



- Split labeled data $N = N_t + N_b$
- Average of predictions on test data with bias correction

$$\bar{O}_{BC} = \frac{1}{M} \sum_{i \in \text{Unlabeled}} O_i^P + \frac{1}{N_b} \sum_{i \in BC} (O_i - O_i^P)$$

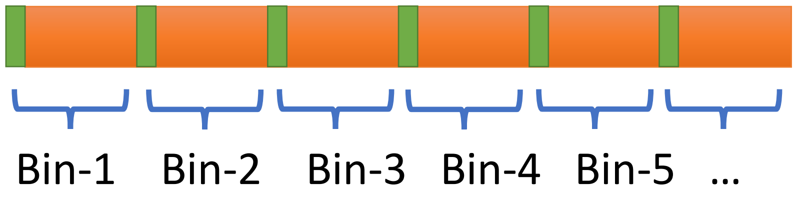
- Expectation value, $\langle \bar{O}_{BC} \rangle = \langle O^P \rangle + \langle O - O^P \rangle = \langle O \rangle$
- BC term converts systematic error of prediction to statistical uncertainty

Incorporating Labeled Data

- Include directly measured values O_i from labeled data

$$\bar{O}_{BC}^{\text{imp}} = w_1 \times \left(\frac{1}{N} \sum_{i \in \text{Labeled}} O_i \right) + w_2 \times \left(\frac{1}{M} \sum_{i \in \text{Unlabeled}} O_i^P + \frac{1}{N_b} \sum_{i \in BC} (O_i - O_i^P) \right)$$

- w_1, w_2: weights determined based on the (co)variance of two terms
- If you need more than just a simple average in data analysis
  - two different data, O_i on labeled and O_i^P on unlabeled samples
  - simultaneous fit on these two data sets with the same fit parameters
  - O_i and O_i^P have the same mean after BC but may have different variance
- Statistical errors can be estimated using Bootstrap resampling
- Binning and BC for each bin is another option for complicated data analysis



Quality of Prediction

- Bias-corrected average

$$\bar{O}_{BC} = \frac{1}{M}\sum_{i\in\text{Unlabeled}} O_i^P + \frac{1}{N_b}\sum_{i\in BC}(O_i - O_i^P)$$

- Statistical error of the unbiased average

$$\sigma^2_{\bar{O}_{BC}} \approx \frac{1}{M}\sigma^2_{O^P} + \frac{1}{N_{bc}}\sigma^2_{O-O^P}$$
$$\approx \frac{\sigma^2_O}{M}\left(1 + \frac{M}{N_{bc}}\frac{\sigma^2_{O-O^P}}{\sigma^2_O}\right) \equiv \frac{\sigma^2_O}{M}\left(1 + \frac{M}{N_{bc}}Q^2\right); \qquad Q^2 \equiv \frac{\sigma^2_{O-O^P}}{\sigma^2_O}$$

approximations ($\approx$) for small correlation between the two terms and a good prediction algorithm that gives $\sigma^2_O \approx \sigma^2_{O^P}$

$$\frac{\sigma^2_{\bar{O}_{BC}}}{\sigma^2_{\bar{O}}} \approx 1 + \frac{M}{N_{bc}}Q^2; \qquad \frac{\sigma_{\bar{O}_{BC}}}{\sigma_{\bar{O}}} \approx 1 + \frac{M}{2N_{bc}}Q^2$$

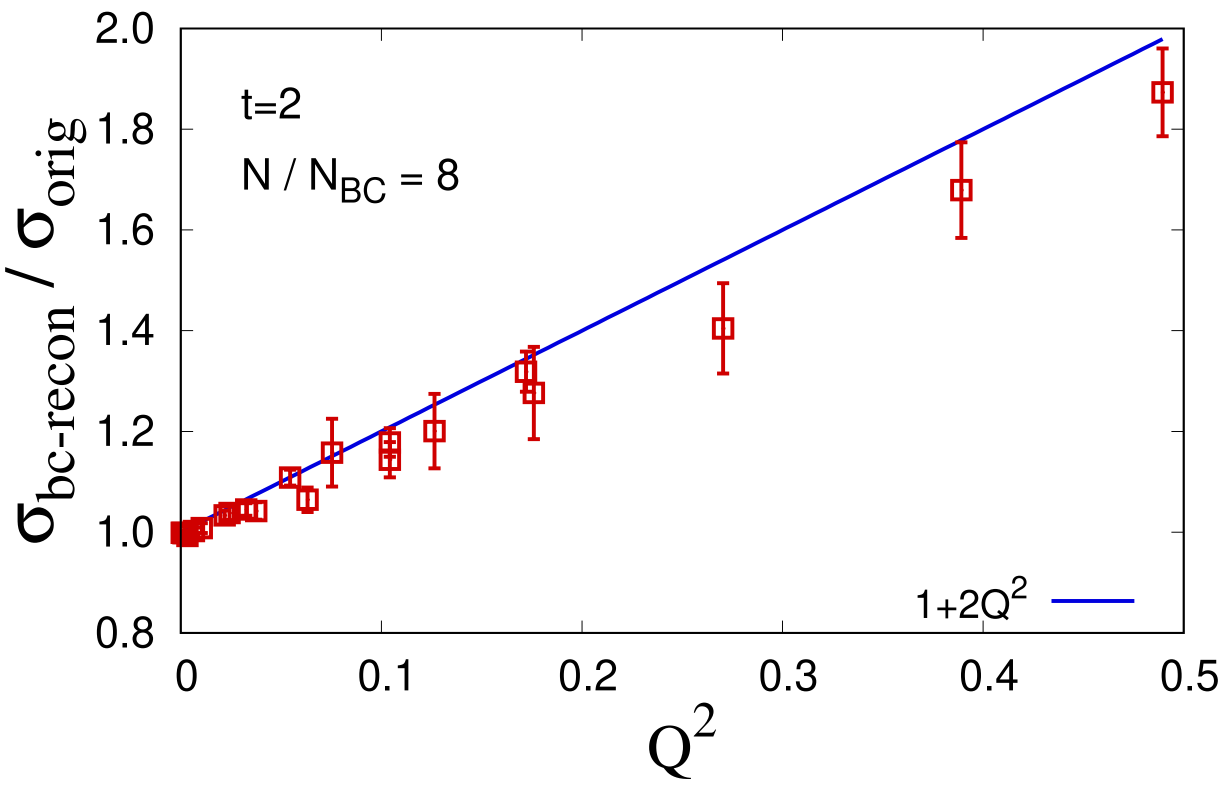
for $N_{bc}/M = 0.2$

Q	Error Increase
0.5	62.5%
0.3	22.5%
0.1	2.5%

- Q-value shows the expected error-increase due to the ML prediction error
- In practice, BC data have less autocorrelation than full data, because of the many measurements per configuration, so $\sigma_{\bar{O}_{BC}}$ gives smaller error than expected above

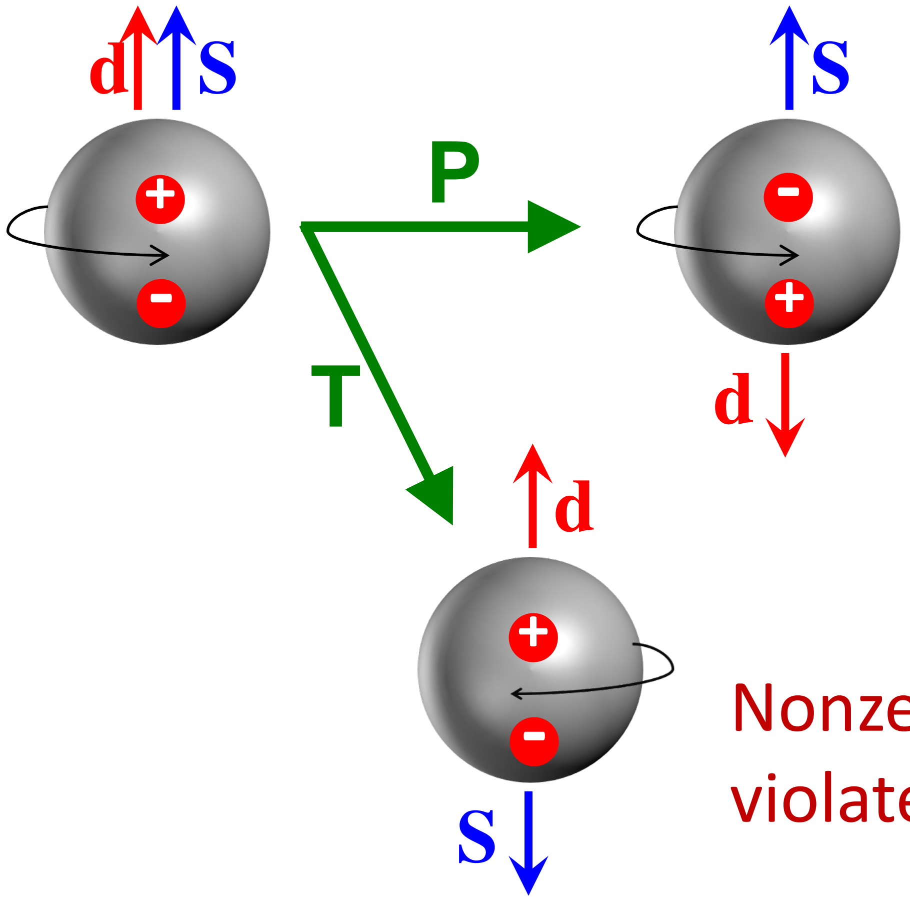
Statistical Error Increase for Different Q-values

- The statistical error increase is proportional to $Q^2 \equiv \frac{\sigma^2_{O-O^P}}{\sigma^2_O}$
- For *independent data*, the error increase ratio due to bias correction is expected to be $1 + \frac{\mathrm{M}}{2\mathrm{N_{bc}}}\mathrm{Q}^2$
- *Correlation between the data samples* makes it $1 + \alpha \frac{\mathrm{M}}{2\mathrm{N_{bc}}}\mathrm{Q}^2$ with $0 < \alpha < 1$

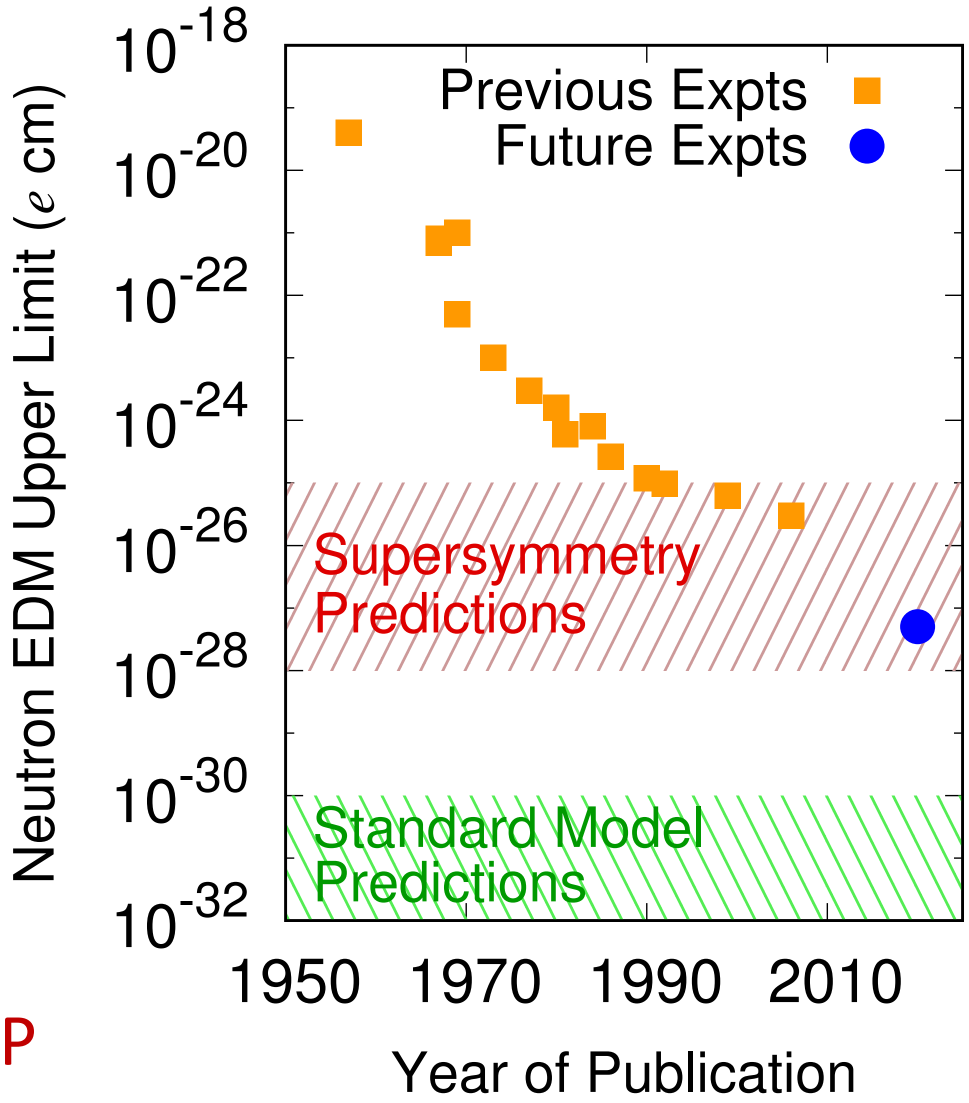


Neutron EDM and CP Violation

- Measures separation between centers of (+) and (-) charges



Nonzero nEDM violates P and T, so CP



Effective CPV Lagrangian

$$\mathcal{L}_{\mathrm{CPV}}^{d\leq 6} = -\frac{g_s^2}{32\pi^2}\bar{\theta}G\tilde{G}$$ dim=4 QCD θ-term

$$-\frac{i}{2}\sum_{q=u,d,s} d_q \bar{q}(\sigma\cdot F)\gamma_5 q$$ dim=5 Quark EDM (qEDM)

$$-\frac{i}{2}\sum_{q=u,d,s} \tilde{d}_q g_s \bar{q}(\sigma\cdot G)\gamma_5 q$$ dim=5 Quark Chromo EDM (CEDM)

$$+d_w \frac{g_s}{6} G\tilde{G}G$$ dim=6 Weinberg 3g operator

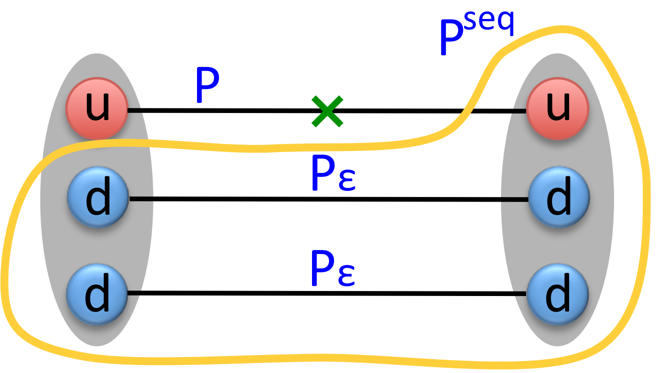
$$+\sum_i C_i^{(4q)} O_i^{(4q)}$$ dim=6 Four-quark operators

Quark Chromo EDM (cEDM)

- Simulation in presence of CPV cEDM interaction

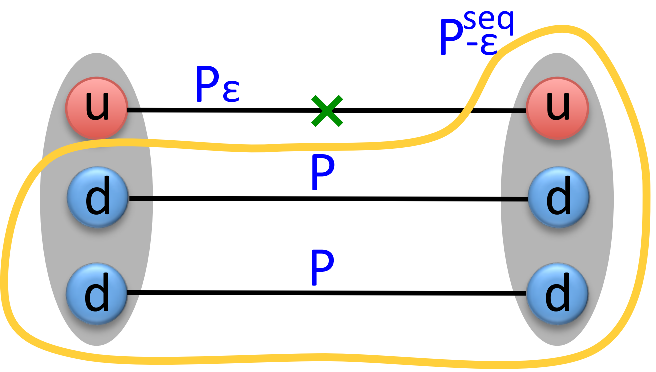
$$S = S_{QCD} + S_{cEDM}$$

$$S_{cEDM} = -\frac{i}{2}\int d^4x \;\; \tilde{d}_q g_s \bar{q}(\sigma \cdot G)\gamma_5 q$$



- Schwinger source method
  Include cEDM term in valence quark propagators
  by modifying Dirac operator

$$D_{\rm clov} \rightarrow D_{\rm clov} + i\varepsilon\sigma^{\mu\nu}\gamma_5 G_{\mu\nu}$$



- cEDM contribution to nEDM can be obtained by calculating vector form-factor F_3 with propagators including cEDM & $O_{\gamma_5} = \bar{q}\gamma_5 q$

Prediction of $C_{2pt}^{\mathbf{CPV}}$ from C_{2pt}

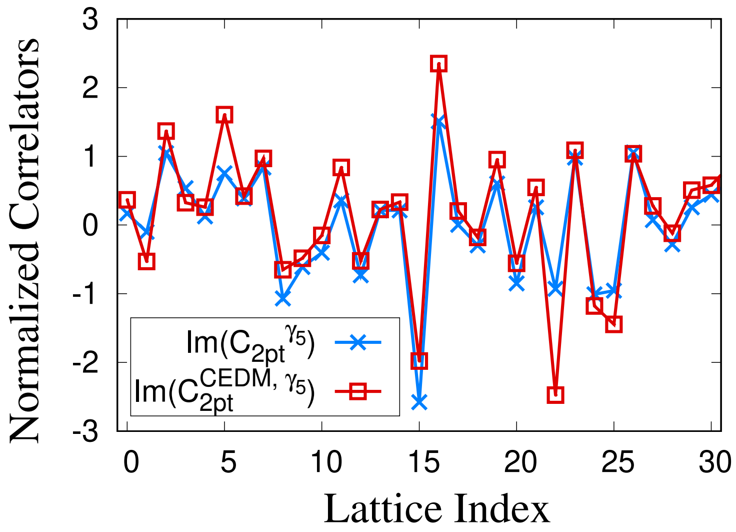
- Predict C_{2pt} for cEDM and γ_5 insertions from C_{2pt} without CPV

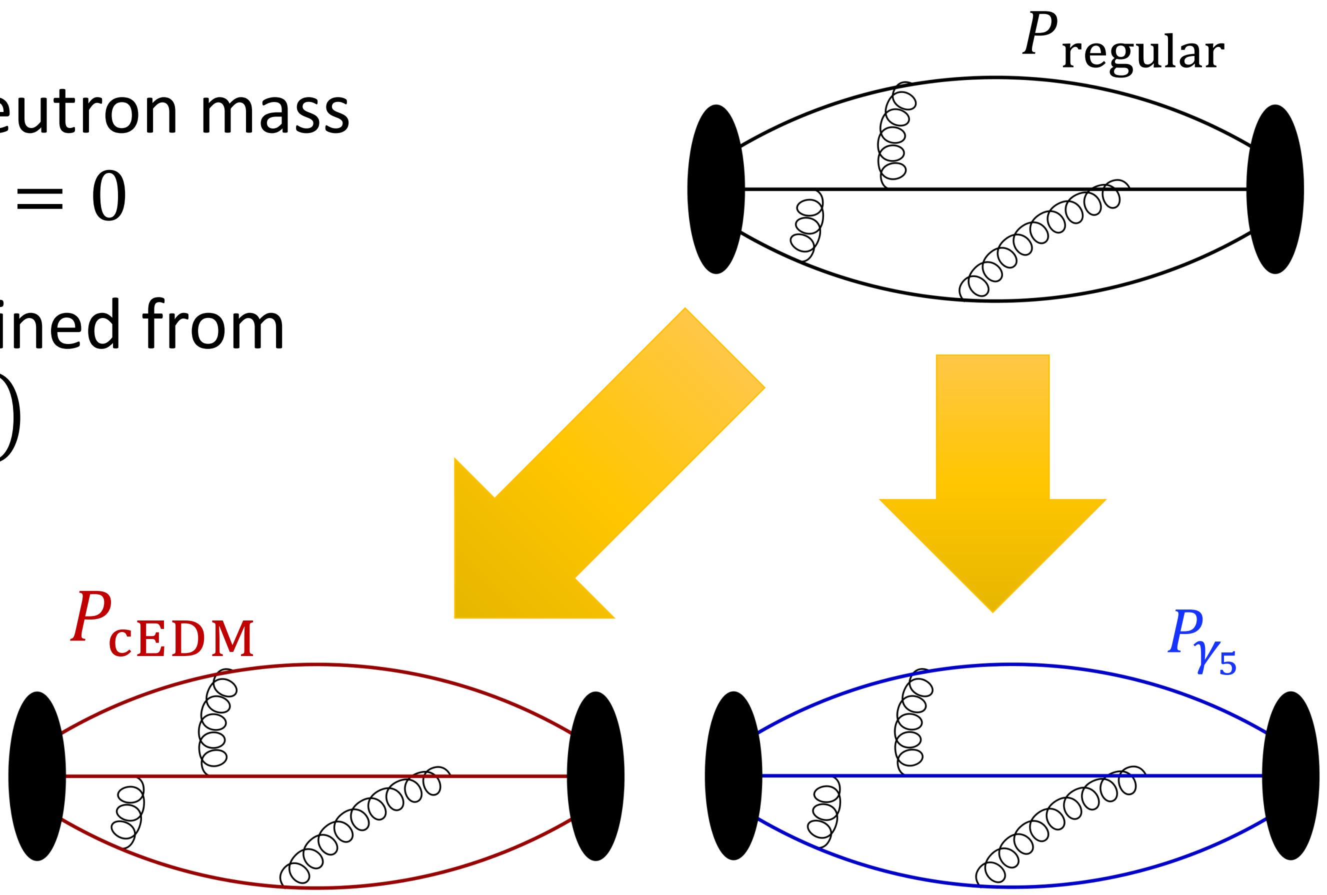
- CPV interactions ➔ phase in neutron mass

$$\left(ip_\mu\gamma_\mu + me^{-2i\alpha\gamma_5}\right)u_N = 0$$

- At leading order, α can be obtained from

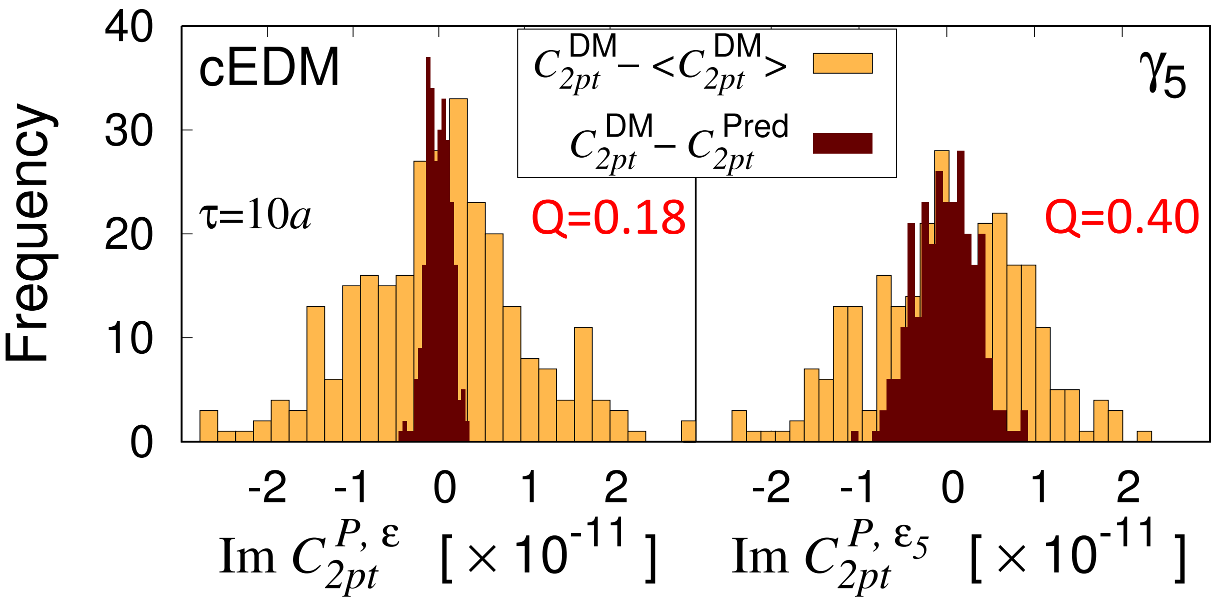
$$C_{2pt}^{P} \equiv \mathrm{Tr}\left(\gamma_5\langle NN^\dagger\rangle\right)$$





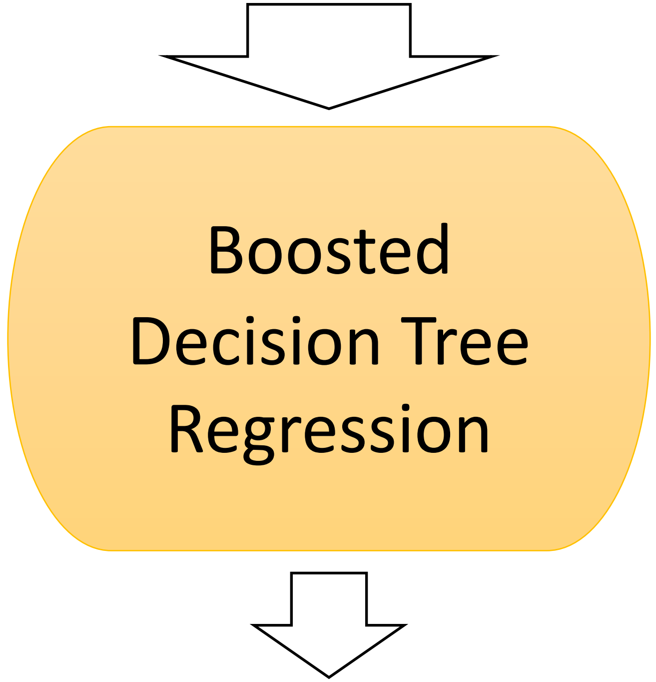
Prediction of C_{2pt}^{CPV} from C_{2pt}

- Training and Test performed for
  - $a = 0.12$ fm, $M_\pi = 305$ MeV
  - Measurements: 400 confs × 64 srcs
- # of training data: 70 confs
  # of BC data: 50 confs
  # of unlabeled data: 280 confs



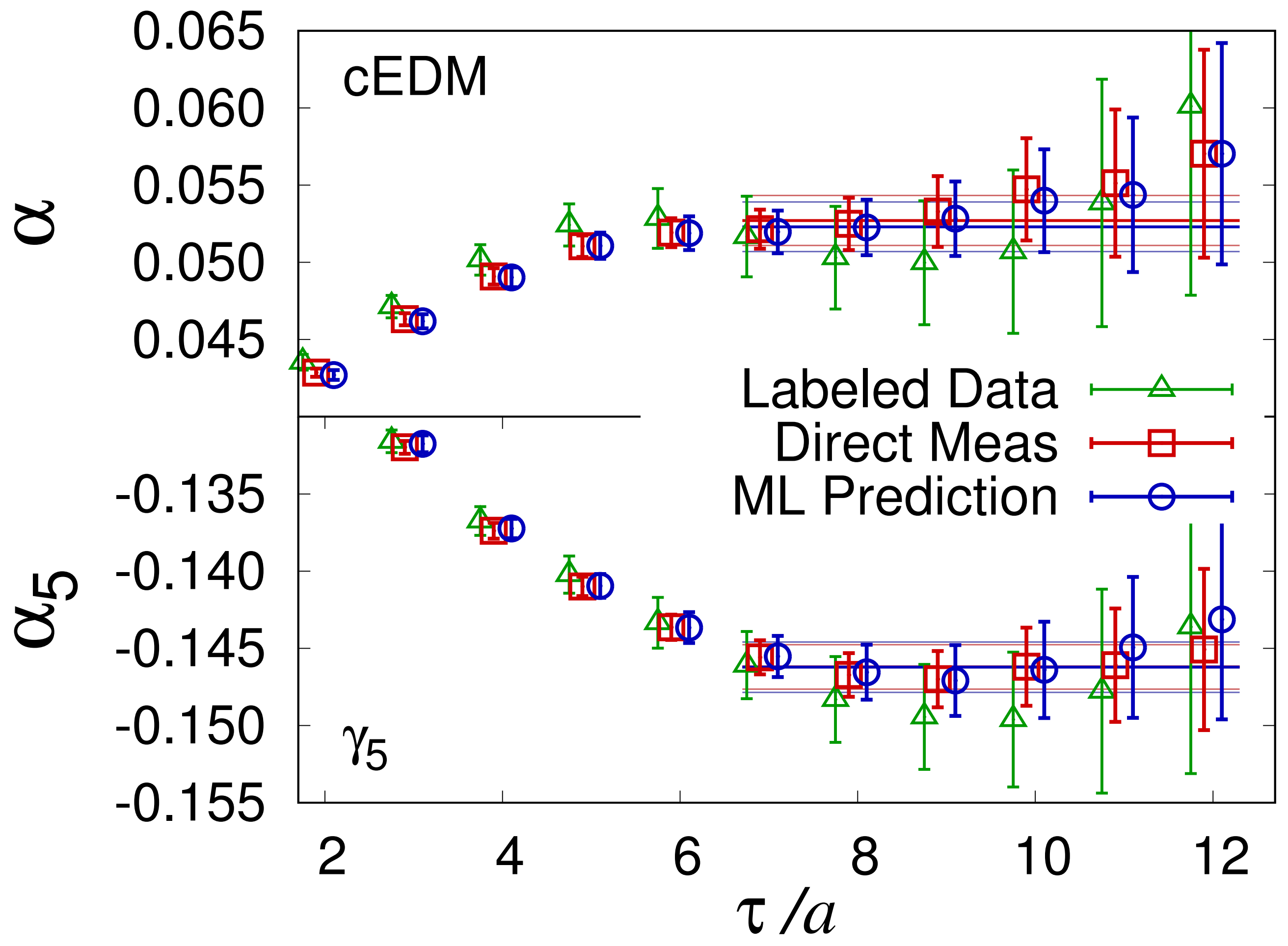
Input:

$$X_i = \{\mathrm{Re}, \mathrm{Im}[C_{2pt}^{S,P}(0 \le \tau/a \le 16)]\}$$



Output: $\mathrm{Im}\left[C_{2pt}^{P\,(\mathrm{cEDM},\gamma_5)}(\tau)\right]$

Prediction of C_{2pt}^{CPV} from C_{2pt}



- $\boldsymbol{\alpha}$ **(cEDM)**
  - DM: 0.0527(17)
  - Prediction: 0.0525(18)
- $\boldsymbol{\alpha_5}$ $(\boldsymbol{\gamma_5})$
  - DM: -0.1463(14)
  - Prediction: -0.1460(17)

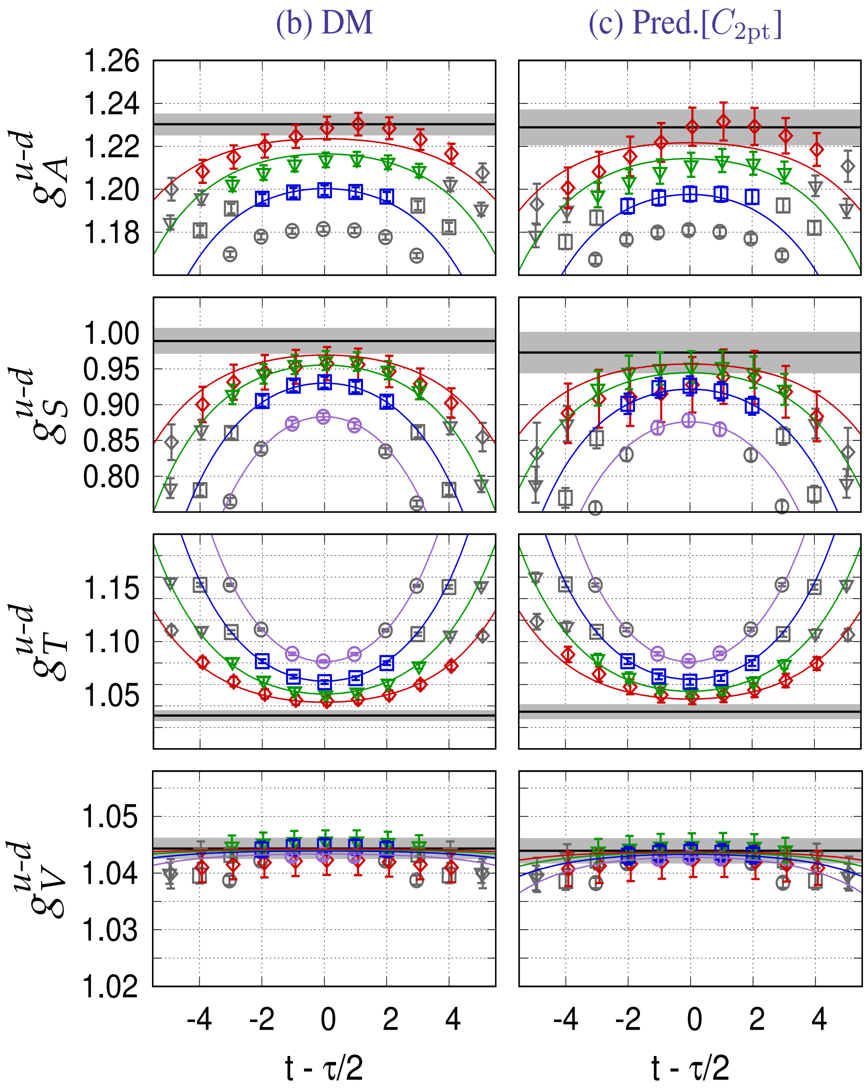
➢ DM: DM on 400 confs

➢ Prediction: DM on 120 confs + ML prediction on 280 confs

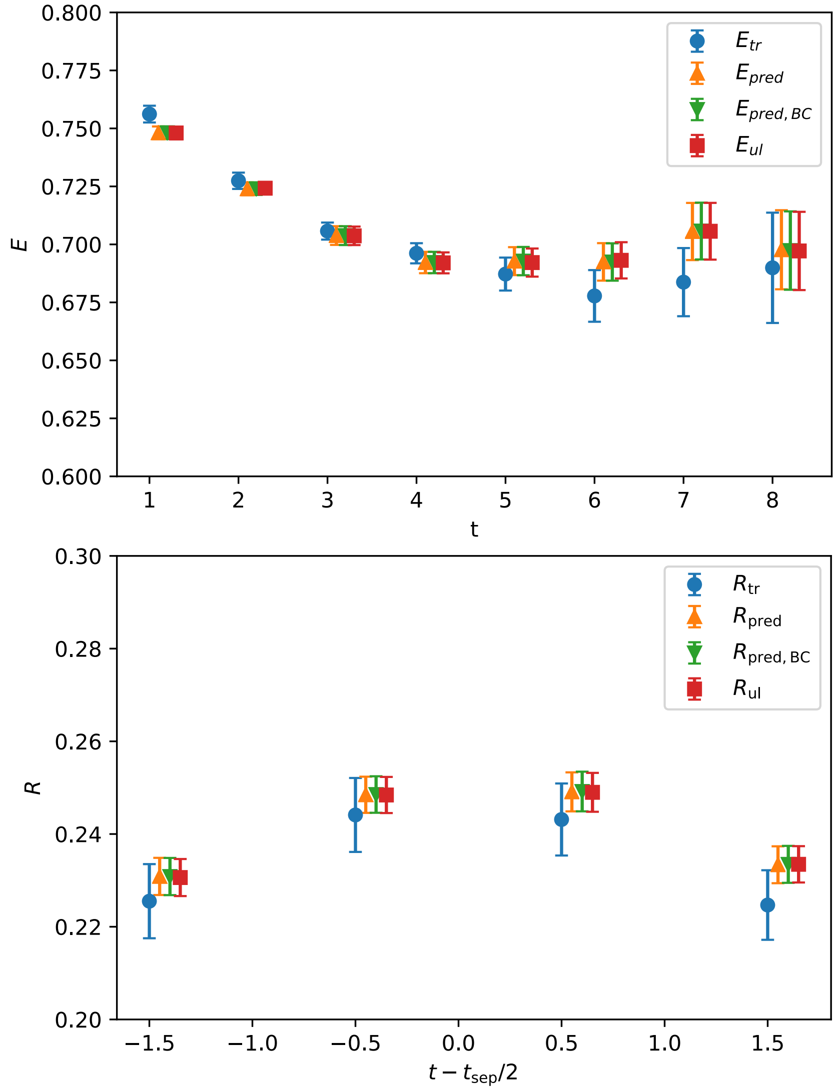
BY, Tanmoy Bhattacharya, Rajan Gupta, PRD 100, 014504 (2019)

Other Applications

BY, Tanmoy Bhattacharya, Rajan Gupta, PRD 100, 014504 (2019)
Rui Zhang, Zhouyou Fan, Ruizi Li, Huey-Wen Lin, BY, PRD 101, 034516 (2020)



Prediction of $\boldsymbol{C_{3pt}}$ from $\boldsymbol{C_{2pt}}$



Prediction of η_s distribution amplitude (upper) and Kaon quasi-PDF (lower) $z = 4$ from $z < 4$

ML Regression Algorithm using D-Wave Quantum Annelaer

ML Regression using D-Wave Quantum Annealer

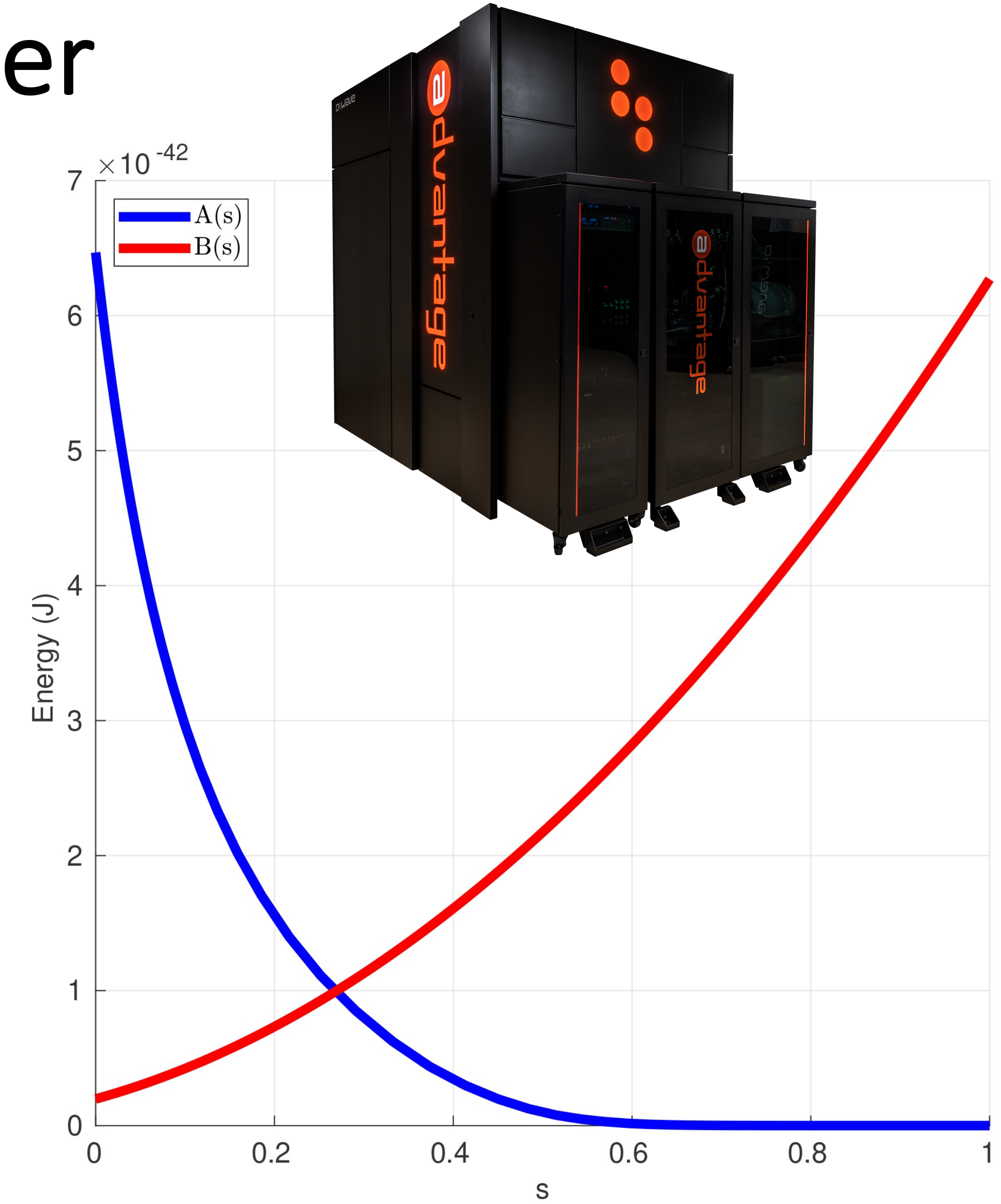
- Most ML algorithms involve optimization problems; many of them rely on stochastic approaches, but expensive for large problems
- D-Wave quantum annealer can be used as a fast and accurate optimizer for ML optimization problems

D-Wave Quantum Annealer

- Hamiltonian

$$H = -\frac{A(s)}{2}\left(\sum_i \hat{\sigma}_x^{(i)}\right) + \frac{B(s)}{2}\left(\sum_i h_i \hat{\sigma}_z^{(i)} + \sum_{i>j} J_{i,j} \hat{\sigma}_z^{(i)} \sigma_z^{(j)}\right)$$

- $h_i, J_{i,j}$: biases and coupling strengths that user can set to their problem parameters
- After annealing at < 15 mK, QPU returns low-energy solution (spin up/down of quantum bits) of the Ising model Hamiltonian
- Large number of reads is required to obtain minimum energy solution for large problems, but each read takes $O(10)\mu s$
- ML typically needs only near-optimal solution



Annealing Time

Sparse Coding

$$\min_{\Phi} \sum_{k=1}^{K} \min_{\vec{a}^{(k)}} \left[\frac{1}{2} \left\|\vec{X}^{(k)} - \boldsymbol{\Phi}\vec{a}^{(k)}\right\|_2 + \lambda \left\|\vec{a}^{(k)}\right\|_0\right]$$

- Unsupervised ML algorithm
- Find dictionary $\boldsymbol{\Phi} \in \mathbb{R}^{D \times N_q}$ and sparse representation $\vec{a}^{(k)} \in \mathbb{R}^{N_q}$ from which input data $\vec{X}^{(k)} \in \mathbb{R}^D$ can be reconstructed by

$$\vec{X}^{(k)} \approx \boldsymbol{\Phi}\vec{a}^{(k)} = a_1^{(k)}\vec{v}_1 + a_2^{(k)}\vec{v}_2 + \cdots + a_1^{(k)}\vec{v}_1$$

- The representation is sparse because the λ-term enforces a minimal set of dictionary elements for the reconstruction of a given input data
- Optimization in $\vec{a}^{(k)}$ of l^0-norm function is a highly non-convex problem

Sparse Coding on D-Wave quantum annealer

$$\min_{\Phi} \sum_{k=1}^{K} \min_{\vec{a}^{(k)}} \left[\frac{1}{2} \left\| \vec{X}^{(k)} - \boldsymbol{\Phi} \vec{a}^{(k)} \right\|_2 + \lambda \left\| \vec{a}^{(k)} \right\|_0 \right]$$

- The sparse coding problem can be mapped onto D-Wave by

$$H(\vec{h}, \boldsymbol{Q}, \vec{a}) = \sum_{i} a_i h_i + \sum_{i<j} Q_{ij} a_i a_j$$

$$\vec{h} = -\boldsymbol{\Phi}^T \vec{X} + \left(\lambda + \frac{1}{2}\right), \qquad \boldsymbol{Q} = \frac{1}{2} \Phi^T \Phi$$

- On D-Wave, a_i is restricted to binary: $\vec{a}^{(k)} \in \{0,1\}^{N_q}$
- D-Wave finds $\vec{a}^{(k)}$ minimizing H
- Optimization for Φ is performed offline (on classical computers)

Inpainting

Nvidia AI Playground - Inpainting

Ground Truth

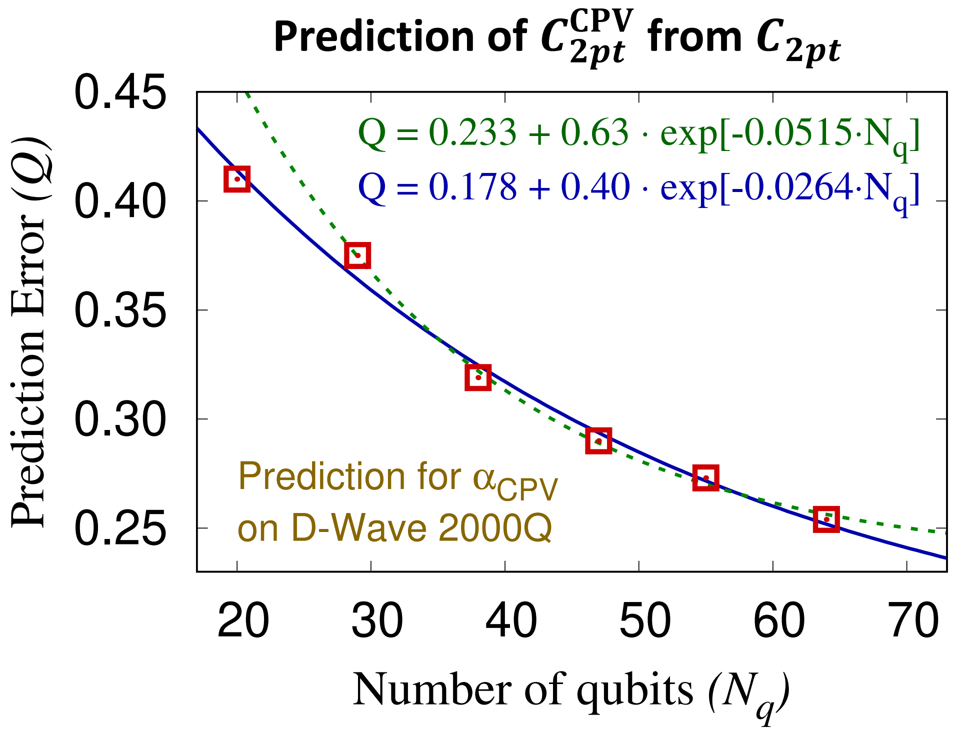
Data with Missing Pixels

Inpainted Results

- Inpainting: restorative conservation where damaged, deteriorating, or missing parts of an artwork are reconstructed as it was originally created
- Sparse coding works as an inpainting algorithm because the reconstruction $\vec{X}^{(k)} \approx \mathbf{\Phi}\vec{a}^{(k)}$ fills up the missing pixels based on the correlation pattern $\mathbf{\Phi}$ learned

Sparse Coding Regression on D-Wave

- Goal: prediction of y from $\vec{x} = \{x_1, x_2, \ldots, x_D\}$
- Procedure:
  1) Obtain $\mathbf{\Phi_0} \in \mathbb{R}^{D \times N_q}$ of $\vec{x}$ from unlabeled data
  2) Extend $\mathbf{\Phi_0}$ to $\mathbf{\Phi} \in \mathbb{R}^{(D+1) \times N_q}$ and encode correlation between $\vec{x}$ and y in Φ using augmented vector $\{\vec{x}, y\}$
  3) For unknown y, reconstruct new vector $\{\vec{x}, \bar{y}\}$ using Φ; reconstruction replaces $\bar{y}$ with its prediction
- This approach is a semi-supervised learning as it utilizes unlabeled data to improve prediction
- D-Wave is used for optimization in $\vec{a}^{(k)}$



- Currently, the performance is limited by the maximum number of qubits available on D-Wave, but the predictions applied on lattice QCD data look promising

Nga Nguyen, Garrett Kenyon, BY, Sci. Rep. 10, 10915 (2020)

Lossy Data Compression Algorithm for Lattice QCD Data

Lossy Data Compression for Lattice QCD

- Modern lattice QCD simulations produce O(PetaBytes) of data that need to be stored for future analysis
- Exploiting correlation between the data components can reduce storage requirement → **Machine learning**
- Reconstruction error sufficiently smaller than the observables statistical fluctuation is good enough for most of the analysis → **Lossy compression**

Lossy Data Compression Algorithm

- **Goal:** find $\mathbf{\Phi} \in \mathbb{R}^{D\times N_q}$ and $\vec{a}^{(k)} \in \{0,1\}^{N_q}$ precisely reconstructing input vectors $\vec{X}^{(k)} \in \mathbb{R}^D$ such that $\vec{X}^{(k)} \approx \mathbf{\Phi}\vec{a}^{(k)} \equiv \vec{X}'^{(k)}$
  - ➢ $\mathbf{\Phi}$ is common for all $k = 1,2,3,\ldots,N$, so memory usage is small
  - ➢ Each vector $\vec{a}^{(k)}$ can be stored in N_q bits
  - ➢ Storing $\left(\{\vec{a}^{(k)}\}_{k=1}^{N}, \mathbf{\Phi}\right)$ for $\{\vec{X}^{(k)}\}_{k=1}^{N}$: compression of D floating-point numbers into N_q bits
  - ➢ Correlation between X_i, encoded in $\mathbf{\Phi}$, allows precise reconstruction with $N_q \ll 32D$
- Such solutions of $\mathbf{\Phi}$ and $\vec{a}^{(k)}$ can be obtained by solving

$$\min_{\mathbf{\Phi}} \sum_{k=1}^{N} \min_{\vec{a}^{(k)}} \left[\left(\vec{X}^{(k)} - \mathbf{\Phi}\vec{a}^{(k)}\right)^2\right]$$

  - ➢ **Finding binary solution of $\vec{a}^{(k)}$ is an NP-hard problem but can be solved using D-Wave**
  - ➢ Finding $\mathbf{\Phi}$ is done on classical computers with stochastic optimizer
  - ➢ Iterate $\vec{a}^{(k)}$- and $\mathbf{\Phi}$-optimizations until it reaches the minimum reconstruction error
  - ➢ Need standardization of $\vec{X}^{(k)}$ beforehand if the data exhibits heteroskedasticity

Bias Correction of Lossy Reconstruction

- Lossy reconstruction introduces error $\vec{X}^{(k)} \neq \mathbf{\Phi}\vec{a}^{(k)} \equiv \vec{X}'^{(k)}$
  Simple average is a biased estimator $\langle f(\vec{X}) \rangle \neq \frac{1}{N}\sum_k f\left(\vec{X}'^{(k)}\right)$
- Unbiased estimator of $\langle f(\vec{X}) \rangle$ can be defined using small portion of original data

$$\bar{\mathrm{O}} = \frac{1}{N}\sum_{k=1}^{N} f\left(\vec{X}'^{(k)}\right) + \frac{1}{N_{bc}}\sum_{k=1}^{N_{bc}}\left(f\left(\vec{X}^{(k)}\right) - f\left(\vec{X}'^{(k)}\right)\right)$$

- Quality of lossy-compression on statistical data

$$\mathrm{Q}^2 \equiv \frac{1}{D}\sum_{i=1}^{D}\frac{\sigma^2_{X_i - X'_i}}{\sigma^2_{X_i}}$$

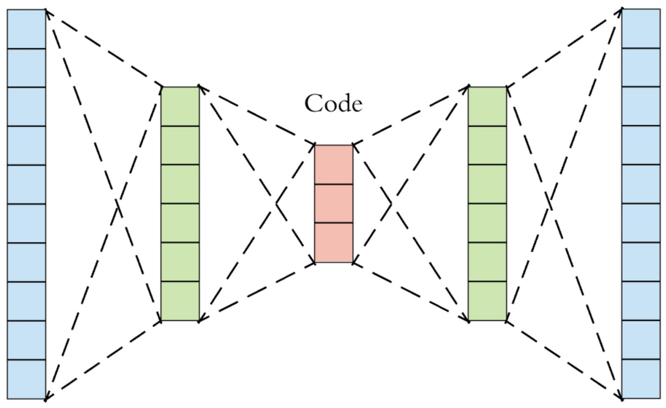
  - Smaller Q^2 indicates the better compression
  - Increase of statistical error due to bias correction is proportional to $\frac{\mathrm{N}}{2\mathrm{N}_{\mathrm{bc}}}\mathrm{Q}^2$
  - eg) With 10% of bias correction data (N_{bc}/N=0.1) and compression of $\mathrm{Q}^2 = 0.01$, original data is typically reconstructed within 5% statical error increase

Comparison with other Algorithms

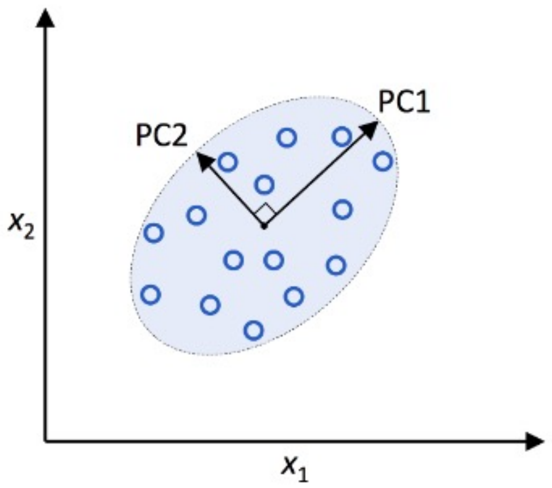
- **Binary compression using D-Wave**
  - Find a set of vectors (Φ) and their binary coefficients ($\boldsymbol{a}^{(k)}$) reconstructing $\boldsymbol{X}^{(k)}$

$$\min_{\Phi} \sum_{k} \min_{\boldsymbol{a}^{(k)}} \left[\sum_{i} \left(X_i^{(k)} - \left[\Phi \boldsymbol{a}^{(k)} \right]_i \right)^2 \right]$$

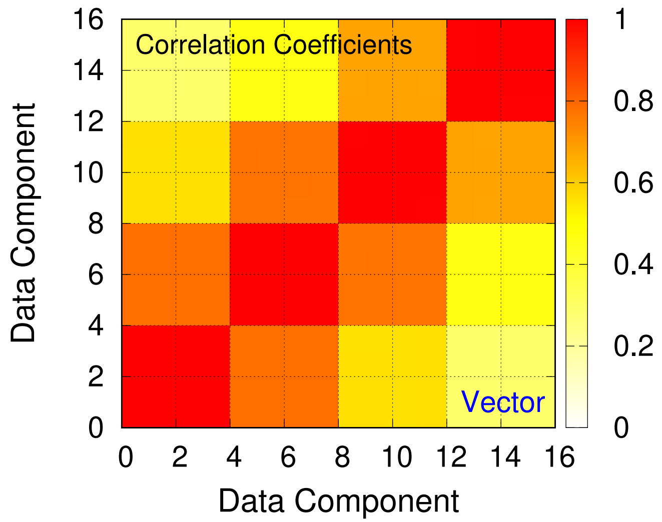
- **Bottle-neck Autoencoder (AE)**
  - Fully connected NN with ReLU
  - Encoder: (16, 128, 64, 32, N_z)
  - Decoder: (N_z, 32, 64, 128, 16)

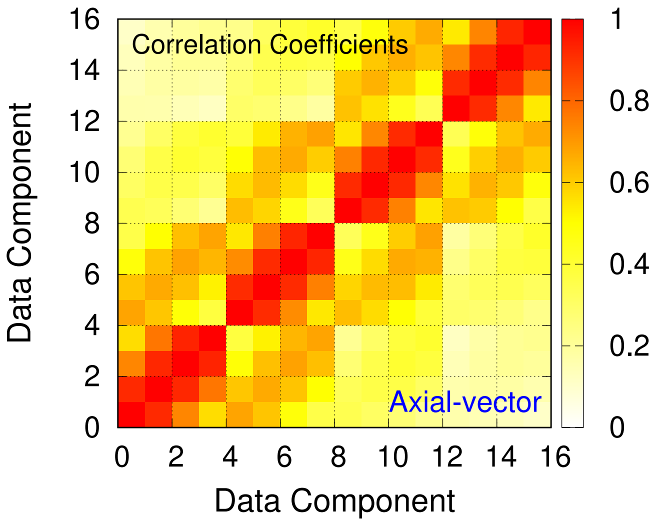


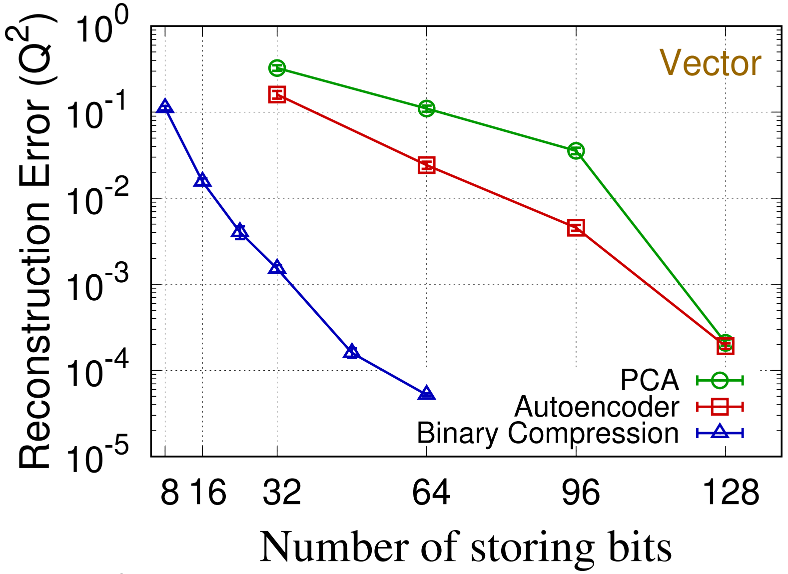
- **Principal Component Analysis (PCA)**
  - Compression by saving the first N_z coefficients of the principal components

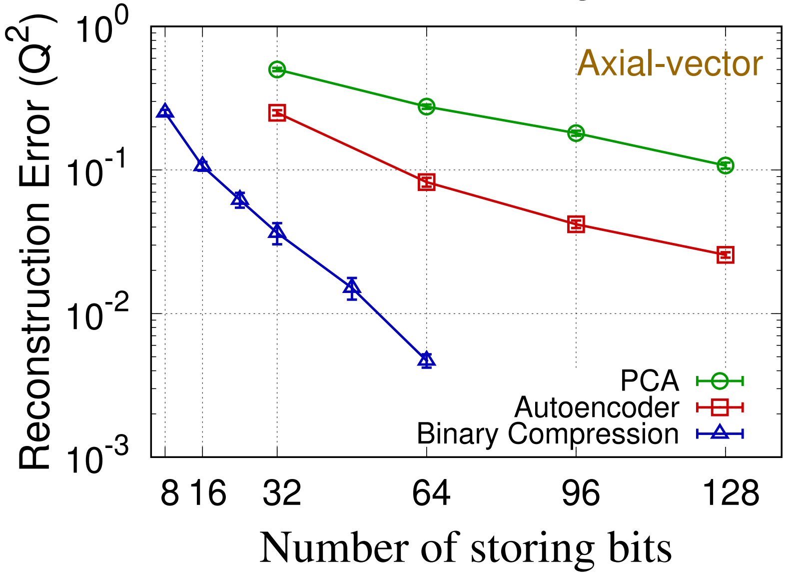


Compression of Lattice QCD data







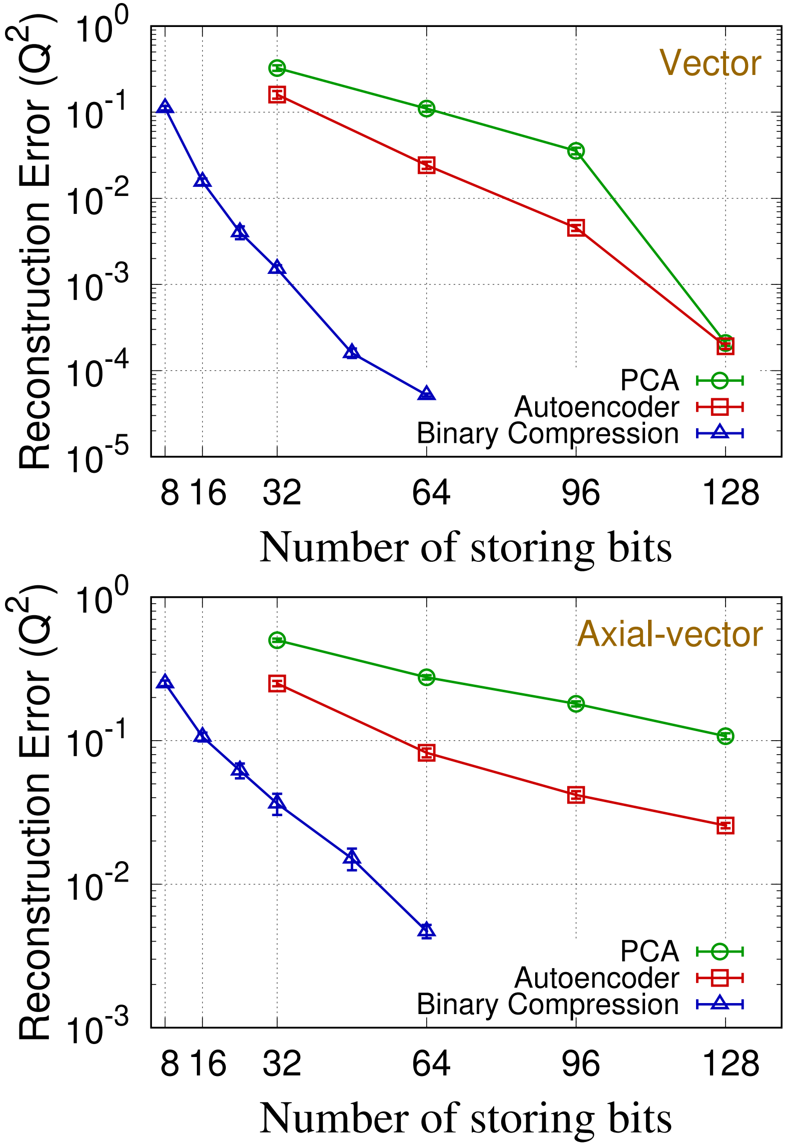


- Compression of "*4 timeslices* X *4 src-sink separations*" of vector and axial-vector nucleon 3pt correlators
- Compression performance of the new algorithm outperforms those based on principal component analysis (PCA) or neural-network autoencoder
- Results from D-Wave simulated annealing; real QPU gives worse performance due to noise in *h* and *J* parameters
- PCA and NN-Autoencoder with single-precision (32bits) codes

BY, Nga Nguyen, Jason Chang, Chia Cheng Chang, Ermal Rrapaj, will appear on arXiv soon

Estimated Error Increase

- $\frac{\sigma_{\text{bc-recon}}}{\sigma_{\text{orig}}} = 1 + \alpha \frac{N}{2N_{\text{bc}}} Q^2$ with $0 < \alpha < 1$
- With 10% of bias correction data ($N/N_{bc} = 10$) and $\alpha = 0.5$, expected error increase is $1 + 2.5Q^2$
- When $Q^2 = 10^{-2}$, expected error increase is 2.5%
- When $Q^2 = 10^{-3}$, expected error increase is 0.25%
- For good lossy compression algorithms, error increase due to bias correction is negligibly small



More Use of Binary Compression Algorithm

- **Outlier detection**
  - An input data with large reconstruction error can be marked anomalous
  - Could find events of new physics or data corruption
- **Cheaper operations in $\vec{a}$-space ($\vec{X}^{(k)} \approx \mathbf{\Phi}\vec{a}^{(k)}$)**
  - Operations on floating-point numbers $\vec{X}^{(k)}$ can be replaced by those on single-bit coefficients $\vec{a}^{(k)}$ with much cheaper computational cost
  - eg 1) sum of vectors $$\sum_{k=1}^{N} \mathbf{X}^{(k)} \approx \sum_{k=1}^{N} \boldsymbol{\phi}\boldsymbol{a}^{(k)} = \boldsymbol{\phi}\left(\sum_{k=1}^{N} \boldsymbol{a}^{(k)}\right)$$
  - eg 2) sum of l^2-norm squares

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

Summary

- Machine learning (ML) is employed to predict unmeasured observables from measured observables
  (Expensive lattice QCD calculation → Cheap ML estimators)
- Bias correction is used to quantify the ML prediction error
- Developed a new regression algorithm utilizing quantum annealer and showed promising prediction ability
- Developed a new ML-based compression algorithm using quantum annealer for binary optimization