Estimation of the Lattice QCD Observables using Machine Learning

Boram Yoon
in collaboration with
Tanmoy Bhattacharya, Rajan Gupta, Garrett Kenyon and Nga Nguyen

Los Alamos National Laboratory

arXiv:1807.05971
Outline

1. Lattice QCD

2. Machine learning (ML) on Lattice QCD observables (statistical data)

3. Correction of the Prediction Bias

4. Applications

5. ML regression on D-Wave quantum annealer
Lattice QCD

• **Non-perturbative** approach to understand QCD
• Formulated on **discretized Euclidean space-time**
  - Hypercubic lattice
  - Lattice spacing \( a \)
  - Quark fields placed on sites
  - Gauge fields on the links between sites; \( U_\mu \)
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

\{M_\pi^{(1)}, F_\pi^{(1)}, C_{3pt:A}^{(1)}, C_{3pt:V}^{(1)}, \ldots\}

\{M_\pi^{(7)}, F_\pi^{(7)}, C_{3pt:A}^{(7)}, C_{3pt:V}^{(7)}, \ldots\}

\[\langle O_X \rangle \approx \frac{1}{N} \sum_{n=1}^{N} O_X^{(n)}\]

Markov Chain Monte Carlo Trajectory of Gibbs Samples
Correlation Map of Nucleon Observables

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

\[ C_{3pt}(\tau=10a, t=5a) \]

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

\[ C_{2pt} \sim \langle N(\tau)N^+(0) \rangle \quad C_{3pt}^{A,S,T,V} \sim \langle N(\tau)O(t)N^+(0) \rangle \]
Machine Learning on Lattice QCD Observables
One can consider the machine learning (ML) process as a data fitting.

The machine $F$ has very general fitting functional form with huge number of free parameters.

The free parameters are determined from large number of training data:

$$F(X_i) \approx y_i$$

For example,$X_i$: pixels of a picture

$y_i$: “cat” or “dog”
Machine Learning on Lattice QCD Observables

• Assume $M$ indep. measurements
• Common observables $X_i$ on all $M$
  Target observable $O_i$ on first $N$

1) **Train** machine $F$ to yield $O_i$ from $X_i$
   on the Labeled Data
2) **Predict** $O_i$ of the Unlabeled data from $X_i$
   
   $F(X_i) = O_i^P \approx O_i$

Input: $X_i = (o_i^1, o_i^2, o_i^3, \ldots)$

[Diagram: Labeled and Unlabeled Data with Machine $F$]
Prediction Bias

- \( F(X_i) = O_i^P \approx O_i \)

- Simple average
  \[
  \bar{O} = \frac{1}{M - N} \sum_{i \in \text{Unlabeled}} O_i^P
  \]
  is not correct due to **prediction bias**

- Prediction = TrueAnswer + Noise + Bias

- ML prediction may have bias
  \[
  \langle O_i^P \rangle \neq \langle O_i \rangle
  \]
  Bias = \( \langle O_i^P \rangle - \langle O_i \rangle \)
Bias Correction

• Split labeled data $N = N_t + N_b$

• Average of predictions on test data with bias correction

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

• Expectation value, $\langle \bar{O} \rangle = \langle O_i^P \rangle + \langle O_i - O_i^P \rangle = \langle O_i \rangle$

• BC term converts systematic error of prediction to statistical uncertainty
Incorporating Labeled Data

- Include directly measured values $O_i$ from labeled data

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

- $w_1, w_2$: 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

- Binning and BC for each bin is another option for complicated data analysis
Error Estimation

- Training, bias correction and predictions are made on statistical data
- Final estimates, including the trained ML algorithm, inherit uncertainties
- We use **bootstrap resampling** for error estimation
  - Random choice of bootstrap samples (labeled and unlabeled data set, separately)
  - Split training, BC data in labeled data
  - Train, BC and obtain estimates on each bootstrap sample
Applications
Nucleon Isovector Charges $g_{A,S,T,V}$

$$\langle p | \bar{u} \Gamma d | n \rangle = g_{\Gamma} \bar{\psi}_{p} \Gamma \psi_{n}$$

- $g_{A} = 1.2772(20)$ [Expt]
- $g_{T}$: quark EDM

- On lattice,

$$\frac{C_{3pt}}{C_{2pt}} \rightarrow g_{\Gamma}$$
Prediction of $C_{3pt}$ from $C_{2pt}$

Input: $X_i = \{C_{2pt}(0 \leq \tau/a \leq T_{\text{max}})\}$

Boosted Decision Tree Regression

Output: $C_{3pt}^{A,S,T,V}(\tau, t)$
Decision Tree Regression

Input: \( C_{2pt}(0 \leq \tau/a \leq 20) \)
Output: \( C_{3pt}^A(10, 5) \)

\[
C_{2pt}(10) \leq 1.384 \\
\text{True} \\
C_{2pt}(10) \leq 0.666 \\
\text{True} \\
C_{2pt}(10) \leq 0.316 \\
-0.942 \\
\text{False} \\
C_{2pt}(10) \leq 2.481 \\
\text{False} \\
C_{2pt}(10) \leq 1.796 \\
0.598 \\
\text{False} \\
C_{2pt}(10) \leq 3.793 \\
2.024 \\
\text{False} \\
C_{2pt}(10) \leq 1.047 \\
-0.171 \\
\text{False} \\
C_{2pt}(10) \leq 1.047 \\
0.206 \\
\text{True} \\
C_{2pt}(10) \leq 1.047 \\
0.206 \\
\text{True} \\
C_{2pt}(10) \leq 0.666 \\
-0.171 \\
\text{False} \\
C_{2pt}(10) \leq 2.481 \\
1.175 \\
\text{False} \\
C_{2pt}(10) \leq 2.481 \\
2.024 \\
\text{False} \\
C_{2pt}(10) \leq 3.793 \\
3.887 \\
\]

\( C_{3pt}^A(\tau/a = 10, t/a = 5) \)
Boosted Decision Tree (BDT)

- **Iterative boosting**

  \[
  F_0 = \text{[Simple DT } h_0]\]
  \[
  F_1 = F_0 + \text{[Simple DT } h_1 \text{ that corrects residual error of } F_0]\]
  \[
  F_2 = F_1 + \text{[Simple DT } h_2 \text{ that corrects residual error of } F_1]\]
  \[
  F_3 = F_2 + \text{[Simple DT } h_3 \text{ that corrects residual error of } F_2]\]
  \[
  \ldots
  \]
  \[
  F_n = F_{n-1} + h_n
  \]

  \[
  F(X) = F_{N_{boost}}(X)
  \]

  - Usually, \( F_n = F_{n-1} + \nu \cdot h_n \) with “learning rate” \( \nu \sim 0.1 \)
  - In this study, \( N_{boost} = 100 - 500 \)
Decision Tree $h_0$ for $C_{3pt}^A(10, 5)$

Input: \{${C_{2pt}(0 \leq \tau/a \leq 20)}$\}

Output: $C_{3pt}^A(10, 5)$

- True:
  - $C_{2pt}(10) \leq 1.384$
  - $C_{2pt}(10) \leq 0.666$
  - $C_{2pt}(10) \leq 0.316$
  - $C_{2pt}(10) \leq -0.942$
  - $C_{2pt}(10) \leq -0.553$

- False:
  - $C_{2pt}(10) \leq 2.481$
  - $C_{2pt}(10) \leq 1.047$
  - $C_{2pt}(10) \leq 0.171$
  - $C_{2pt}(10) \leq 0.206$
  - $C_{2pt}(10) \leq 0.598$
  - $C_{2pt}(10) \leq 1.175$
  - $C_{2pt}(10) \leq 2.024$
  - $C_{2pt}(10) \leq 3.887$
Revisit: Correlation Map of Nucleon Observables

$C_{3pt}(\tau=10a, \ t=5a)$

<table>
<thead>
<tr>
<th>Correlation coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
</tr>
<tr>
<td>0.9</td>
</tr>
<tr>
<td>0.8</td>
</tr>
<tr>
<td>0.7</td>
</tr>
<tr>
<td>0.6</td>
</tr>
<tr>
<td>0.5</td>
</tr>
<tr>
<td>0.4</td>
</tr>
<tr>
<td>0.3</td>
</tr>
</tbody>
</table>

$C_{2pt}(\tau/a)$

5 6 7 8 9 10 11 12 13
Decision Tree $h_5$ for $C^A_{3pt}(10, 5)$

Input: $\{C_{2pt}(0 \leq \tau/a \leq 20)\}$
Output: $C^A_{3pt}(10, 5)$

- $C_{2pt}(9) \leq 1.386$
  - True
    - $C_{2pt}(10) \leq 0.698$
      - $C_{2pt}(10) \leq 0.347$
        - $-0.572$
      - $C_{2pt}(10) \leq 1.139$
        - $-0.304$
  - False
    - $C_{2pt}(10) \leq 2.971$
      - $C_{2pt}(10) \leq 2.090$
        - $-0.077$
      - $C_{2pt}(12) \leq 20.52$
        - $0.178$
        - $0.446$
        - $0.934$
        - $1.741$
        - $16.03$
Decision Tree $h_{30}$ for $C_{3pt}^A(10, 5)$

Input: $\{C_{2pt}(0 \leq \tau/a \leq 20)\}$
Output: $C_{3pt}^A(10, 5)$

$C_{2pt}(6) \leq 1.729$

- **True**
  - $C_{2pt}(6) \leq 0.435$
    - $C_{2pt}(14) \leq -1.856$
      - $-0.742$
    - $C_{2pt}(13) \leq 8.078$
      - $-0.098$

- **False**
  - $C_{2pt}(10) \leq 5.315$
    - $C_{2pt}(10) \leq -1.265$
      - $-1.254$
    - $C_{2pt}(6) \leq 2.528$
      - $-2.673$

$C_{3pt}(10, 5)$
Prediction of $C_{3pt}$ from $C_{2pt}$

- Training and Test performed for
  - Clover-on-HISQ
  - $a = 0.089 \text{fm, } M_{\pi} = 313 \text{ MeV}$
  - Measurements: 2263 confs $\times$ 64 srcs

- # of training data: 60 confs
- # of BC data: 620 confs
- # of unlabeled data: 1583 confs

- Prediction error
  $$\text{PE} = C_{3pt}^{\text{Direct Meas.}} - C_{3pt}^{\text{Pred}}$$

- Prediction quality for $C_{3pt}^\Gamma(10,5)$
  $$\frac{\sigma_{\text{PE}}}{\sigma_{DM}} = 0.79 \ (S), \ 0.49 \ (A), \ 0.44 \ (T), \ 0.12 \ (V)$$
Prediction of $C_{3pt}$ from $C_{2pt}$

- Prediction quality $C_{3pt}^{\Gamma}(10,5)$
  \[
  \frac{\sigma_{PE}}{\sigma_{DM}} = 0.79 \, (S), \, 0.49 \, (A), \, 0.44 \, (T), \, 0.12 \, (V)
  \]

- Predictions of $C_{3pt}^{\Gamma}(10,5)/\langle C_{2pt}(10) \rangle$

<table>
<thead>
<tr>
<th>$\Gamma$</th>
<th>DM</th>
<th>BC-Prediction</th>
<th>Raw-Prediction</th>
<th>Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>0.936(10)</td>
<td>0.933(15)</td>
<td>0.931(45)</td>
<td>+0.002(46)</td>
</tr>
<tr>
<td>A</td>
<td>1.2011(41)</td>
<td>1.1997(48)</td>
<td>1.1999(109)</td>
<td>−0.0003(105)</td>
</tr>
<tr>
<td>T</td>
<td>1.0627(34)</td>
<td>1.0638(39)</td>
<td>1.0642(79)</td>
<td>−0.0004(78)</td>
</tr>
<tr>
<td>V</td>
<td>1.0462(36)</td>
<td>1.0455(36)</td>
<td>1.0453(39)</td>
<td>+0.0002(20)</td>
</tr>
</tbody>
</table>
### Prediction of $C_{3pt}$ from $C_{2pt}$

<table>
<thead>
<tr>
<th>$g_{u-d}$</th>
<th>$\delta A$</th>
<th>$g_{u-d}$</th>
<th>$\delta S$</th>
<th>$g_{u-d}$</th>
<th>$\delta T$</th>
<th>$g_{u-d}$</th>
<th>$\delta V$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau = \infty$</td>
<td><img src="a" alt="Graph" /> Labeled Data</td>
<td><img src="b" alt="Graph" /> DM</td>
<td><img src="c" alt="Graph" /> Pred.$[C_{2pt}]$</td>
<td>$\tau = 14$</td>
<td><img src="d" alt="Graph" /> Pred.$[C_{2pt}, C_{3pt}]$ (12)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $g_{u-d}$: Graph demonstrating the variation of $g_{u-d}$ with respect to $\delta A$, $\delta S$, $\delta T$, and $\delta V$ for different values of $\tau$. Each graph contains data points for different values of $\tau$, such as $\tau = 14$, $\tau = 10$, $\tau = 12$, and $\tau = 8$. The graphs illustrate the predicted $C_{3pt}$ from $C_{2pt}$ for various time delays $\tau$. The $x$-axis represents $t - \tau/2$, and the $y$-axis represents $g_{u-d}$, while $\delta A$, $\delta S$, $\delta T$, and $\delta V$ are shown as error bars or shaded regions for labeled data.
Prediction of $C_{3pt}(8, 10, 14)$ from $C_{2pt}$ and $C_{3pt}(12)$

- We need $C_{3pt}$ at four values of source-sink separations ($\tau = 8, 10, 12, 14$) for analysis
- $C_{3pt}(8,10,14)$ can be predicted from $C_{2pt}$ and $C_{3pt}(12)$ data
- Prediction quality $C_{3pt}^\Gamma (10,5)$
  \[ \frac{\sigma_{PE}}{\sigma_{DM}} = 0.53 \text{ (S)}, 0.35 \text{ (A)}, 0.33 \text{ (T)}, 0.10 \text{ (V)} \]
  (c.f.) only from $C_{2pt}$: 0.79 (S), 0.49 (A), 0.44 (T), 0.12 (V)
Prediction of $C_{3pt}$ from $C_{2pt}$ and more

(a) Labeled Data  (b) DM  (c) Pred.[$C_{2pt}$]  (d) Pred.[$C_{2pt},C_{3pt}(12)$]
## Prediction of $C_{3pt}$ from $C_{2pt}$

- Results extrapolated to $\tau \to \infty$

<table>
<thead>
<tr>
<th></th>
<th>DM</th>
<th>Pred.$[C_{2pt}]$</th>
<th>Pred.$[C_{2pt},C_{3pt}(12)]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g_S$</td>
<td>0.989(18)</td>
<td>0.973(29)</td>
<td>0.981(20)</td>
</tr>
<tr>
<td>$g_A$</td>
<td>1.2303(51)</td>
<td>1.2289(83)</td>
<td>1.2304(61)</td>
</tr>
<tr>
<td>$g_T$</td>
<td>1.0311(51)</td>
<td>1.0347(68)</td>
<td>1.0326(54)</td>
</tr>
<tr>
<td>$g_V$</td>
<td>1.0443(19)</td>
<td>1.0439(22)</td>
<td>1.0440(21)</td>
</tr>
</tbody>
</table>

|               | 2263 DM (Direct Meas.) | 680 DM + 1583 Pred. | 680 DM + 1583 Pred. |
Neutron EDM and CP Violation

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

Nonzero nEDM violates P and T, so CP

Neutron EDM Upper Limit (e cm)

Year of Publication

Previous Expts

Future Expts

Supersymmetry Predictions

Standard Model Predictions
Effective CPV Lagrangian

\[ \mathcal{L}^{d \leq 6}_{\text{CPV}} = -\frac{g_s^2}{32\pi^2} \bar{\theta} G \tilde{G} \]

\[ -\frac{i}{2} \sum_{q=u,d,s} d_q \bar{q} (\sigma \cdot F) \gamma_5 q \]

\[ -\frac{i}{2} \sum_{q=u,d,s} \tilde{d}_q g_s \bar{q} (\sigma \cdot G) \gamma_5 q \]

\[ + d_w \frac{g_s}{6} G \tilde{G} G \]

\[ + \sum_i C_i^{(4q)} O_i^{(4q)} \]

- dim=4  QCD $\theta$-term
- dim=5  Quark EDM (qEDM)
- dim=5  Quark Chromo EDM (CEDM)
- dim=6  Weinberg 3g operator
- dim=6  Four-quark operators
Quark Chromo EDM (cEDM)

- Simulation in presence of CPV cEDM interaction
  \[ S = S_{QCD} + S_{cEDM} \]
  \[ S_{cEDM} = -\frac{i}{2} \int d^4 x \, \bar{d}_q g_s q (\sigma \cdot G) \gamma_5 q \]

- Schwinger source method
  Include cEDM term in valence quark propagators by modifying Dirac operator
  \[ D_{\text{clov}} \rightarrow D_{\text{clov}} + i \epsilon \sigma^{\mu\nu} \gamma_5 G_{\mu\nu} \]

- 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 $C_{2pt}^{CPV}$ from $C_{2pt}$

- Predict $C_{2pt}$ for cEDM and $\gamma_5$ insertions from $C_{2pt}$ without CPV.
- CPV interactions $\Rightarrow$ phase in neutron mass
  
  $$
  (i p_\mu \gamma_\mu + me^{-2i\alpha\gamma_5})u_N = 0
  $$

- At leading order, $\alpha$ can be obtained from
  
  $$
  C_{2pt}^P \equiv \text{Tr}(\gamma_5\langle NN^+\rangle)
  $$
Prediction of $C_{2pt}^{CPV}$ from $C_{2pt}$

- Training and Test performed for
  - $a = 0.12$ fm, $M_\pi = 305$ MeV
  - Measurements: 400 confs $\times$ 64 srcs

- # of training data: 70 confs
- # of BC data: 50 confs
- # of unlabeled data: 280 confs

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

Output:
$$\text{Im} \left[ C_{2pt}^P (c\text{EDM}, r_5)(\tau) \right]$$
Prediction of $C_{2pt}^{CPV}$ from $C_{2pt}$

- **$\alpha$ (cEDM)**
  - DM: 0.0527(17)
  - Prediction: 0.0525(18)

- **$\alpha_5 (\gamma_5)$**
  - DM: -0.1463(14)
  - Prediction: -0.1460(17)

- DM: DM on 400 confs
- Prediction: DM on 120 confs + ML prediction on 280 confs
<table>
<thead>
<tr>
<th></th>
<th>Linear Regression</th>
<th>BDT</th>
<th>Neural Network</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Speed</strong></td>
<td>Fastest</td>
<td>Fast</td>
<td>Slow</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>Bad for nonlinear</td>
<td>Okay</td>
<td>Possibly better</td>
</tr>
<tr>
<td><strong>Tuning Parameters</strong></td>
<td>None or a few</td>
<td>Few; not sensitive</td>
<td>Many; sensitive</td>
</tr>
<tr>
<td><strong>Overfitting Risk</strong></td>
<td>Very Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td><strong>Training Data Requirement</strong></td>
<td>Small</td>
<td>Medium</td>
<td>Large</td>
</tr>
<tr>
<td><strong>Interpretability</strong></td>
<td>Yes</td>
<td>Somewhat</td>
<td>Not likely</td>
</tr>
</tbody>
</table>
BDT with scikit-learn Python ML Library

```python
>>> import numpy
>>> from sklearn.ensemble import GradientBoostingRegressor
>>> X = numpy.random.uniform(size=(100,2))*10  # 100 random samples
>>> y = [x[0]**2 + 2*x[1] for x in X]

>>> gbr = GradientBoostingRegressor()
>>> gbr.fit(X,y)  # Training

>>> gbr.predict([[3,4]])  # 3^2+2*4 = 17
array([15.20630936])

>>> gbr.predict([[6,3]])  # 6^2+2*3 = 42
array([42.77231812])

>>> gbr.predict([[8,5]])  # 8^2+2*5 = 74
array([74.14274825])
```
Machine Learning Regression on D-Wave Quantum Annealer
D-Wave Quantum Computer

\[ \mathcal{H}_{\text{ising}} = -\frac{A(s)}{2} \left( \sum_i \hat{\sigma}_x^{(i)} \right) + \frac{B(s)}{2} \left( \sum_i h_i \hat{\sigma}_z^{(i)} + \sum_{i>j} J_{i,j} \hat{\sigma}_z^{(i)} \hat{\sigma}_z^{(j)} \right) \]

- \( \hat{\sigma}_{x,z}^{(i)} \): Pauli matrices operating on a qubit \( q_i \)
- \( h_i, J_{i,j} \): qubit biases and coupling strengths; set by user
- Qubits collapse to classical states (0 or 1), the lowest-energy state of the final H
D-Wave Quantum Computer

• The D-Wave QPU
  - Lattice of interconnected (Chimera) qubits
  - Not fully connected

• Minor Embedding
  - Mapping logical qubits to physical qubits
  - On D-Wave 2000Q (~2048 qubits), about 65 fully connected logical qubits can be mapped via Embedding
Sparse Coding

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

• Unsupervised ML algorithm

• Find dictionary $\Phi$ and sparse representation $\tilde{a}^{(k)}$ from which input data can be reconstructed by

$\tilde{I}^{(k)} \approx \Phi \tilde{a}^{(k)}$

• The representation is sparse because the $\lambda$-term enforces a minimal set of dictionary elements for the reconstruction of a given input data
Sparse coding on D-Wave quantum annealer

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

\[ H(\vec{h}, Q, \vec{a}) = \sum_i a_i h_i + \sum_{\langle i, j \rangle} Q_{ij} a_i a_j \]

\[ \vec{h} = -\Phi^T \vec{1} + \left( \lambda + \frac{1}{2} \right), \quad Q = \frac{1}{2} \Phi^T \Phi \]

• D-Wave finds \( \vec{a}^{(k)} \) minimizing \( H \) (Restriction: \( a_i = 0 \) or 1)

• Optimization for \( \Phi \) is performed offline (on classical computers)
Regression on D-Wave

- Predictions on $C_{2pt}^{CPV}$, using only $C_{2pt}$ calculated without CPV, obtained on D-Wave 2000Q
- After trained with labeled data, reconstruction of an input sample with unknown target observable gives prediction
- Fully connected 64 qubits by embedding
- Promising result: the narrower spread of the prediction error compared to the distribution of original data
Summary

• Machine learning (ML) is employed to predict unmeasured observables from measured observables

• Bias correction and bootstrap error estimation are used

• Demonstrated for two lattice QCD calculations
  1) Prediction of $C_{3pt}$ from $C_{2pt}$
  2) Prediction of $C_{2pt}^{CPV}$ from $C_{2pt}$

• A preliminary study performed on D-Wave 2000Q shows promising results for a regression algorithm.
Acknowledgement

• This work was supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics under Contract No. 89233218CNA000001, and by the LANL LDRD program.

• Computations were carried out using NERSC, OLCF, USQCD Collaboration, LANL IC and D-Wave 2000Q.