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

Tanmoy Bhattacharya

Rajan Gupta

Garrett Kenyon

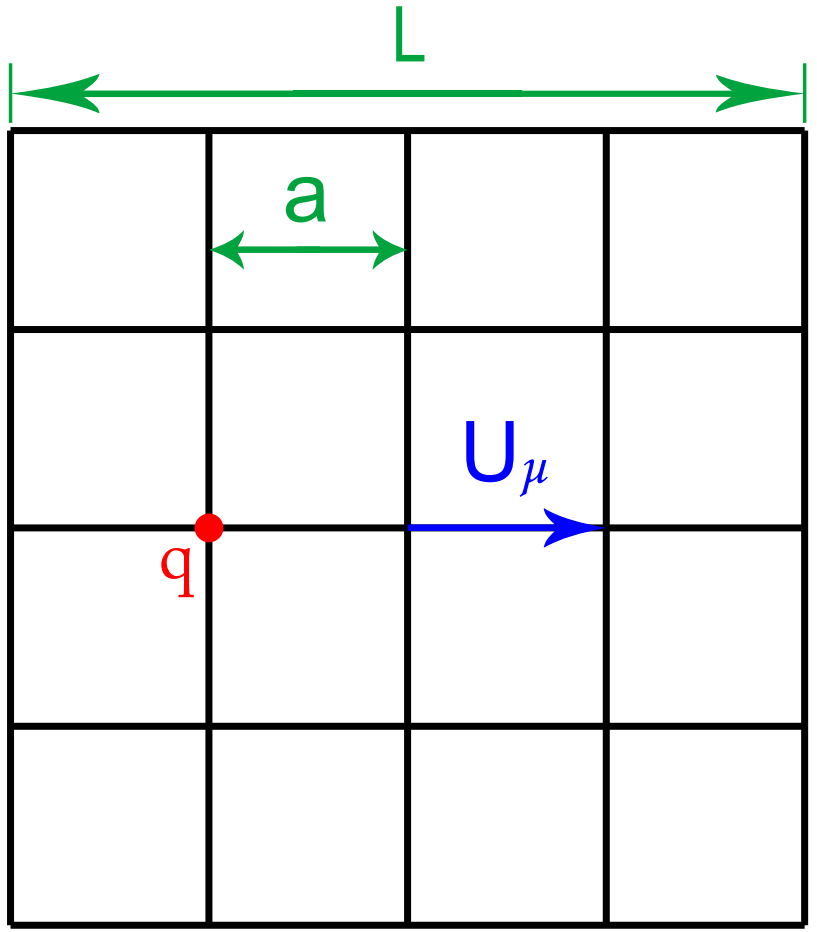
Nga T. T. Nguyen

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_μ



Lattice QCD

- Correlation functions

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

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

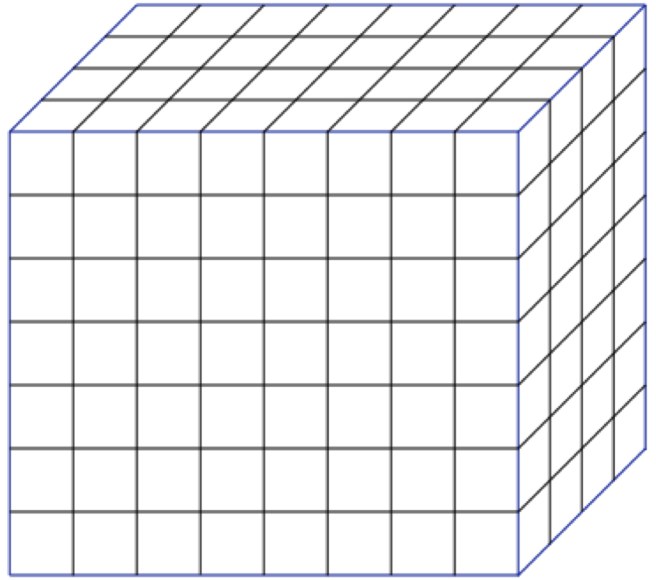
- Monte-Carlo integration
  - Integration variable U is huge

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

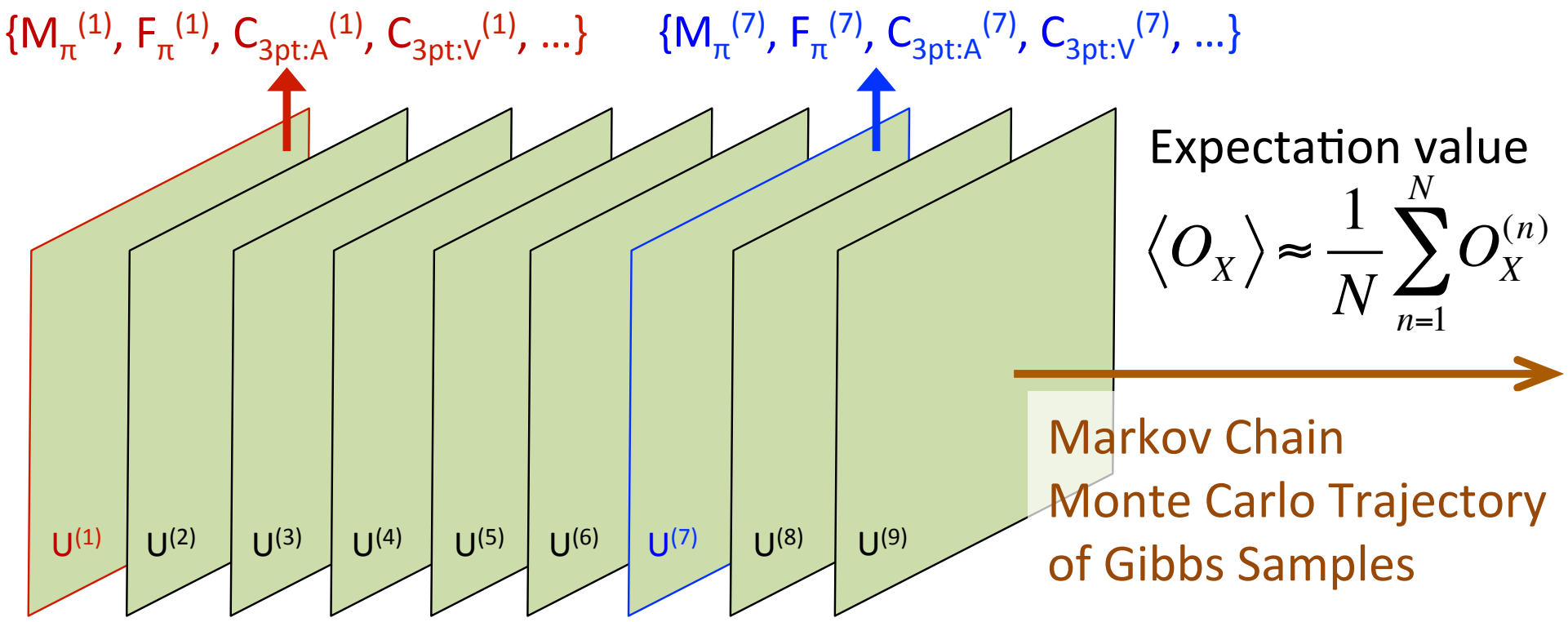
  - Generate Markov chain of gauge configurations U
  - Calculate average as expectation value

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

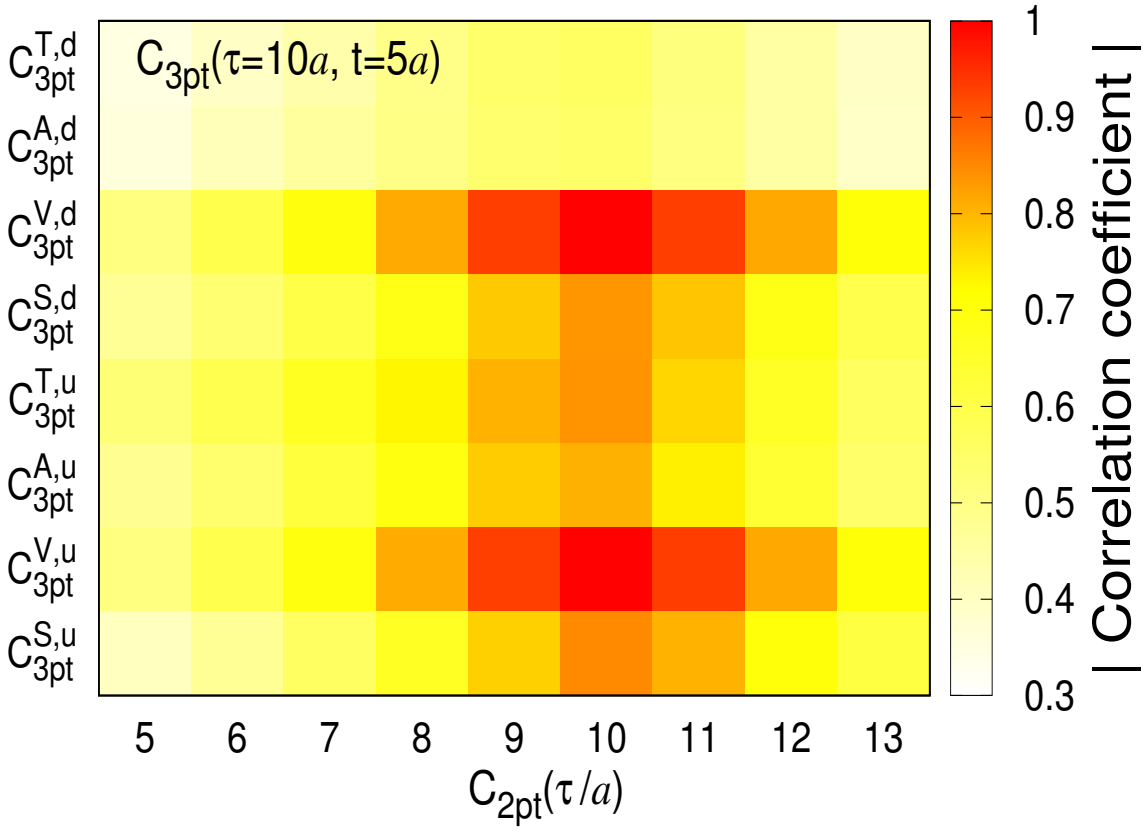
  - Calculation of $O_i \left(U, (D+m_q)^{-1}\right)$: measurement
  - $(D+m)^{-1}$ is computationally expensive



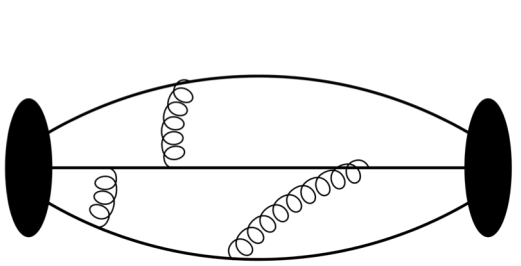
Lattice QCD Observables are Correlated

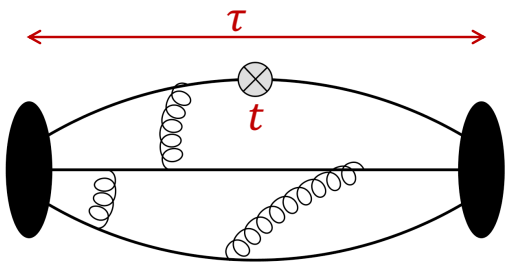


Correlation Map of Nucleon Observables



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





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

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

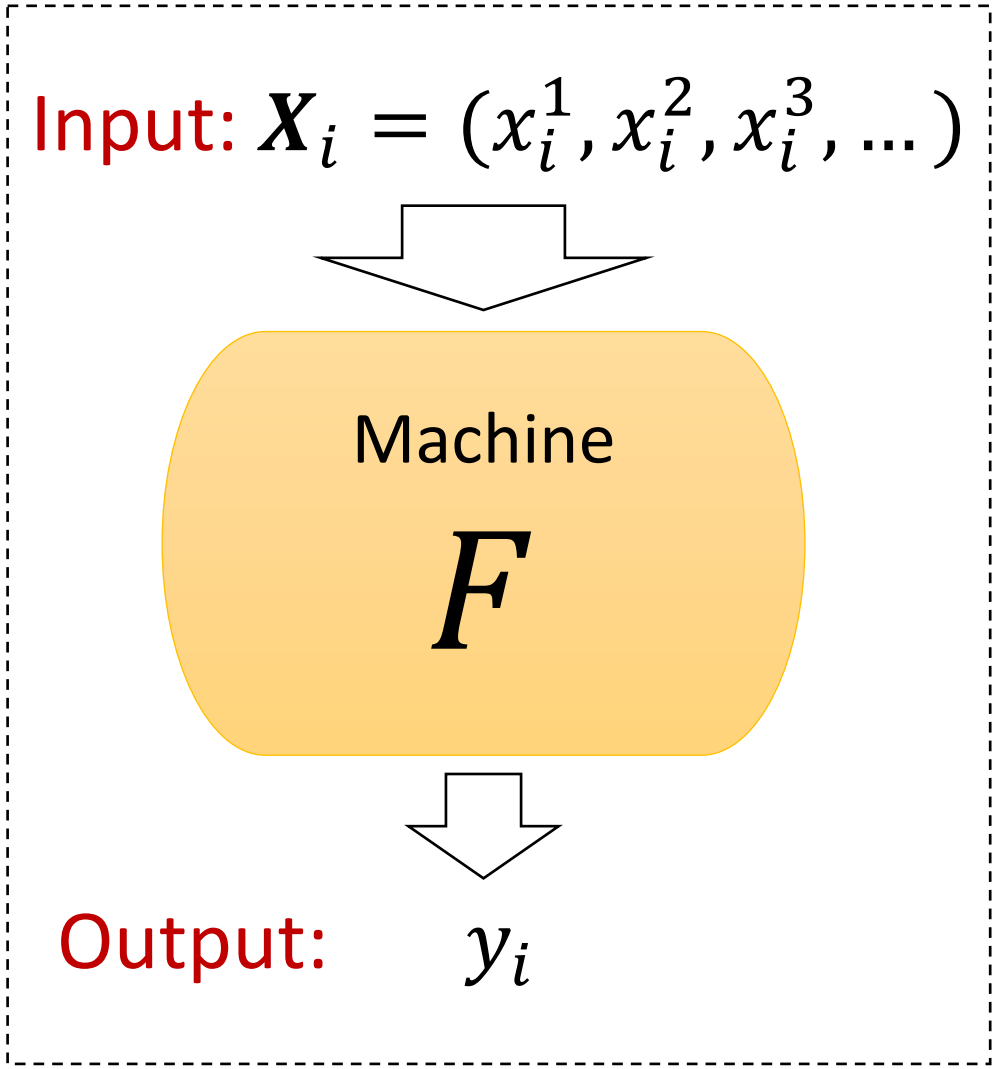
Machine Learning on Lattice QCD Observables

Machine Learning

- 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:

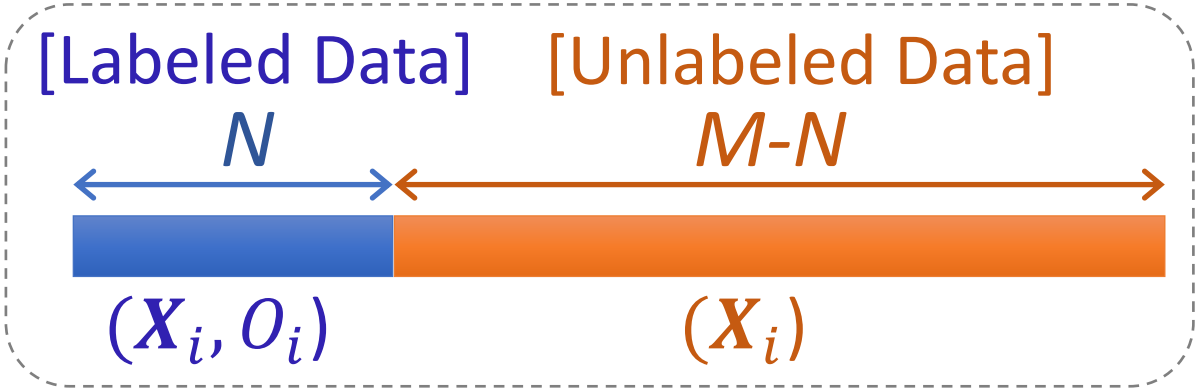
$$F(\boldsymbol{X}_i) \approx y_i$$

- For example,
  $\boldsymbol{X}_i$: pixels of a picture
  y_i : "***cat***" or "***dog***"



Machine Learning on Lattice QCD Observables

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

Machine

$\boldsymbol{F}$

Output: O_i

Prediction Bias

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

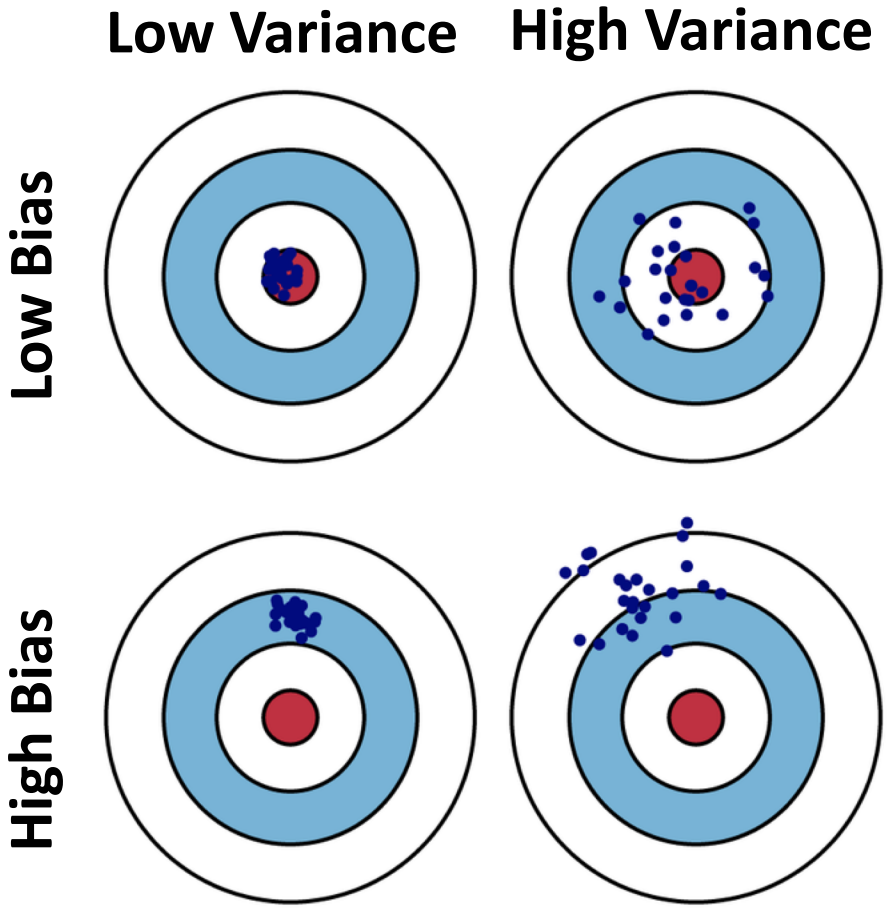
$$\overline{O} = \frac{1}{M-N} \sum_{i \in \text{Unlabeled}} O_i^P$$

  is not correct due to prediction bias

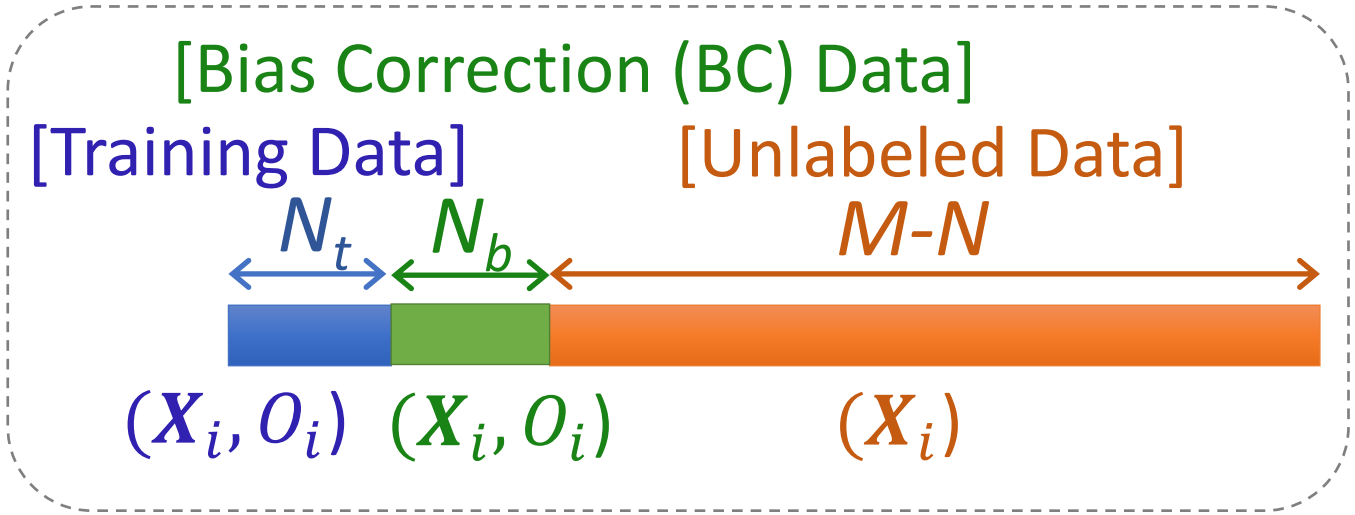
- Prediction = TrueAnswer + Noise + Bias
- ML prediction may have bias

$$\langle O_i^P \rangle \neq \langle O_i \rangle$$

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



Bias Correction



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

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

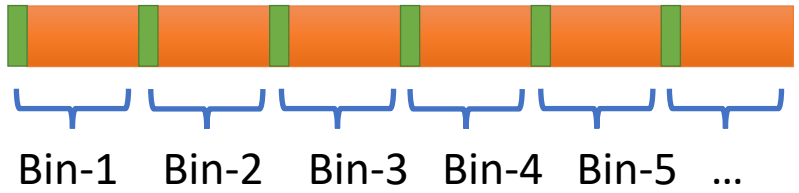
- Expectation value, $\langle \overline{O} \rangle = \langle O_i^P \rangle + \langle O_i - O_i^P \rangle = \langle O_i \rangle$
- BC term converts systematic error of prediction to statistical uncertainty

Incorporating Labeled Data

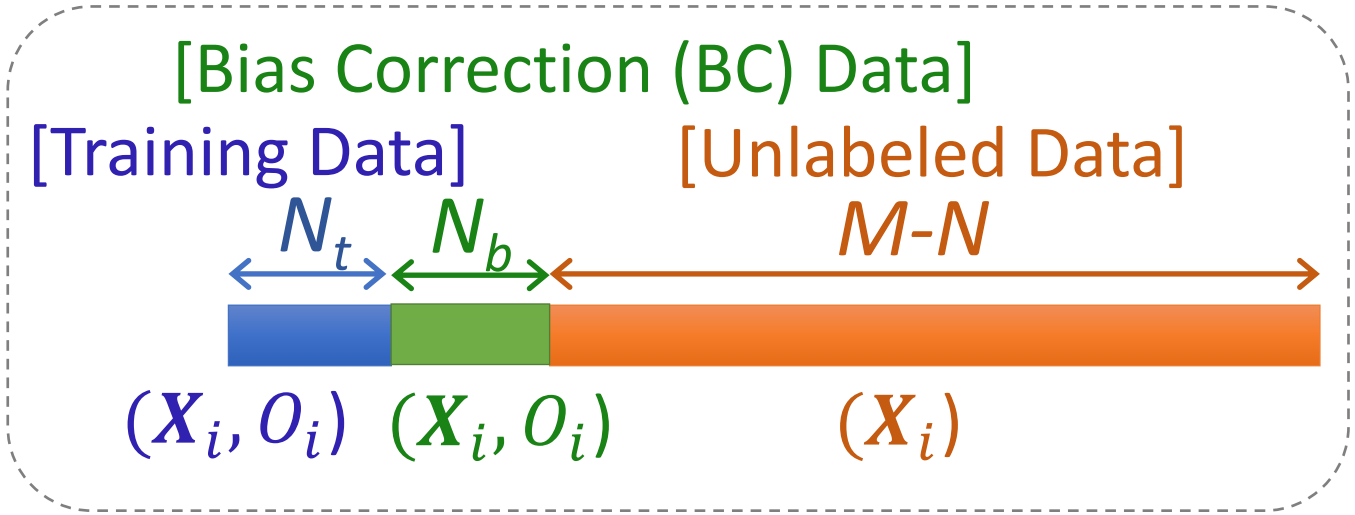
- Include directly measured values O_i from labeled data

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

- 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

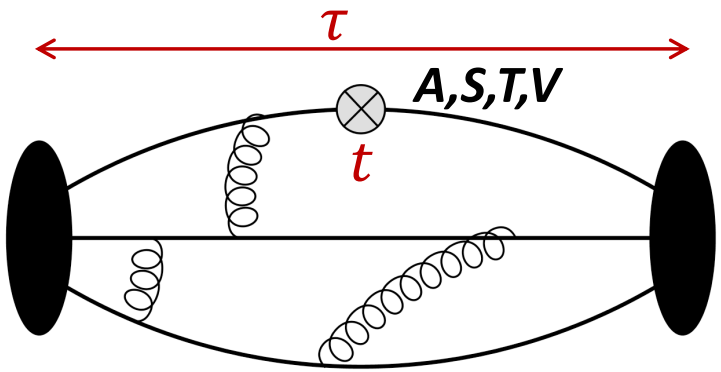
Nucleon Isovector Charges $g_{A,S,T,V}$

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

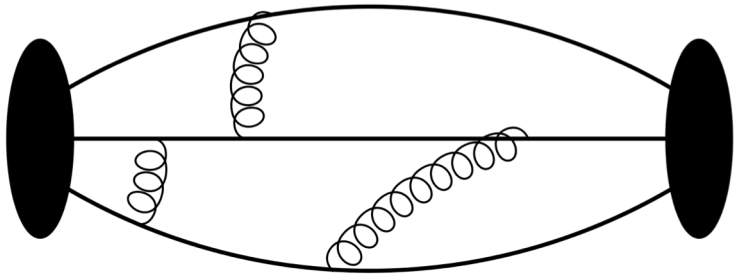
- $g_A = 1.2772(20)$ [Expt]
- g_T: quark EDM

- On lattice,

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



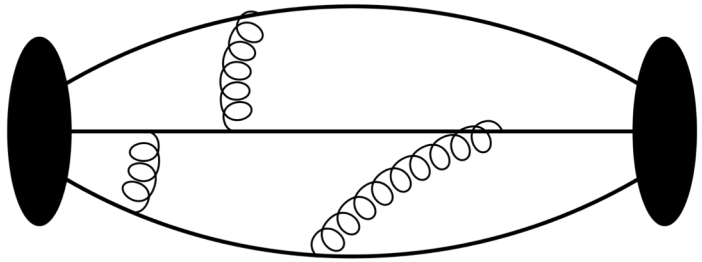
$C_{3pt}(\tau, t)$

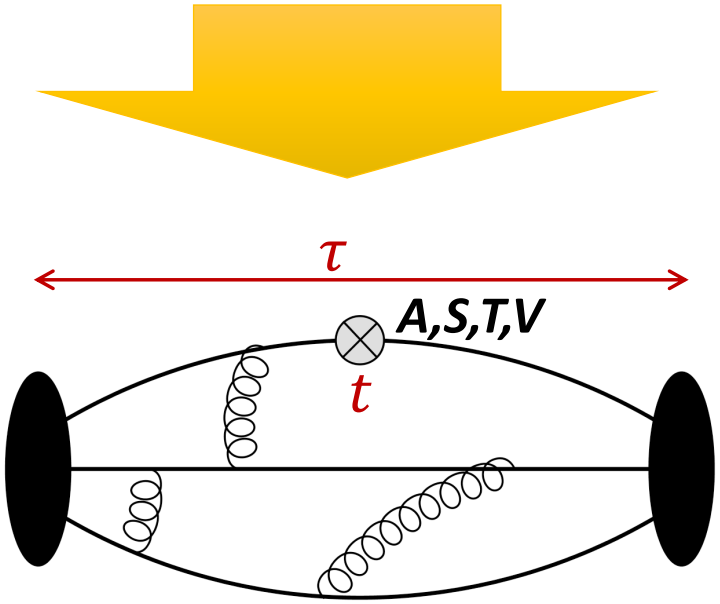


$C_{2pt}(\tau)$

Prediction of C_{3pt} from C_{2pt}

$C_{2pt}(\tau)$





$C_{3pt}(\tau, t)$

Input: $\boldsymbol{X}_i = \{C_{2pt}(0 \leq \tau/a \leq T_{\max})\}$

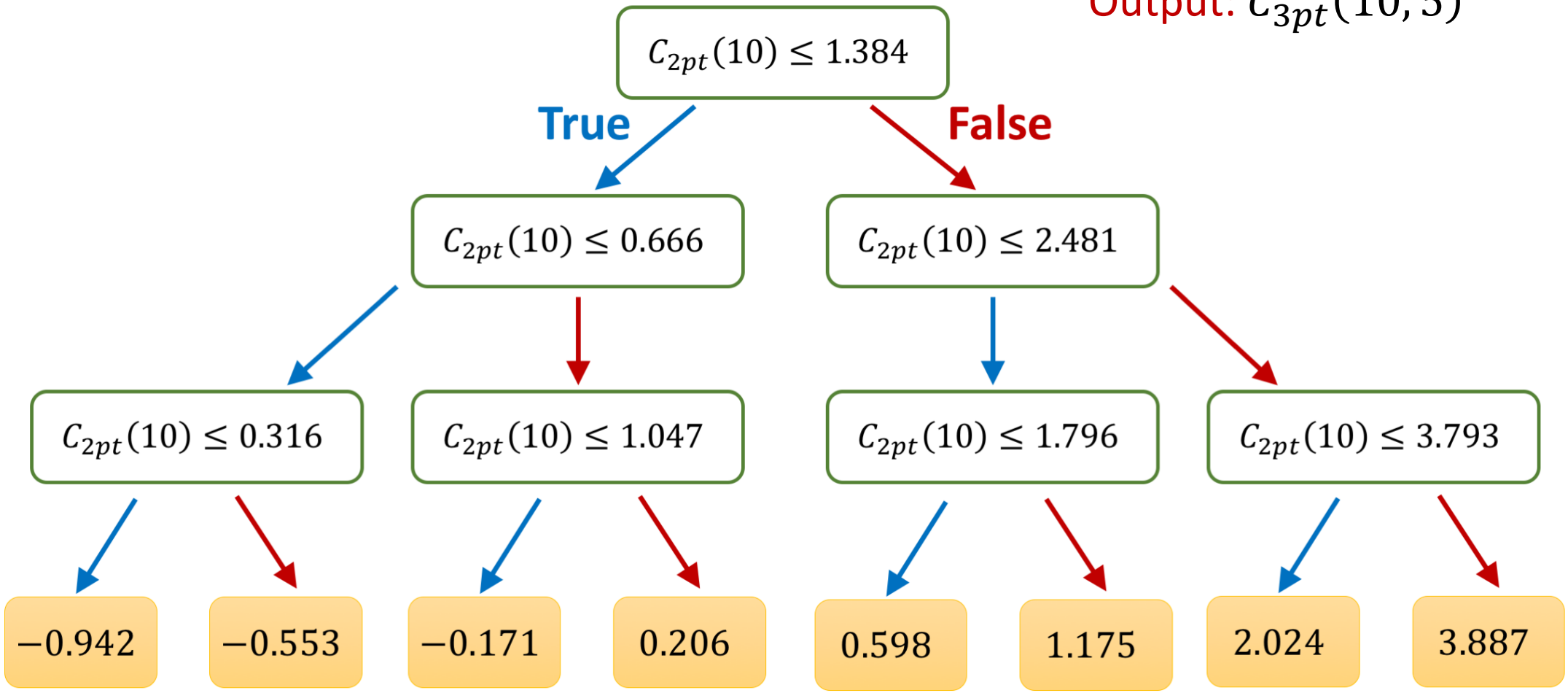
Boosted Decision Tree Regression

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

Decision Tree Regression

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

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



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

Boosted Decision Tree (BDT)

- **Iterative boosting**

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

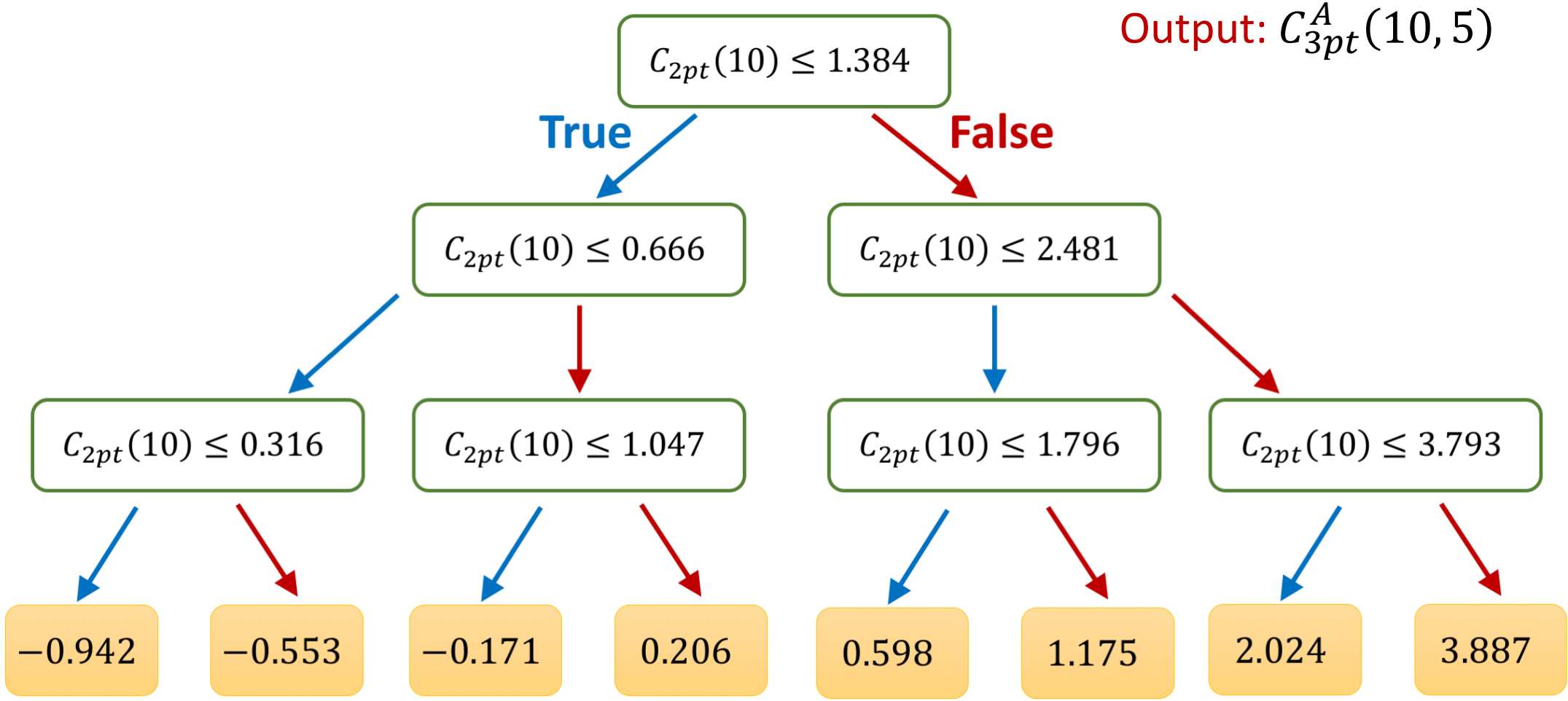
$$F(\boldsymbol{X}) = F_{N_{boost}}(\boldsymbol{X})$$

- Usually, $F_n = F_{n-1} + \nu \cdot h_n$ with "learning rate" $\nu \sim 0.1$
- In this study, $N_{boost} = 100 - 500$

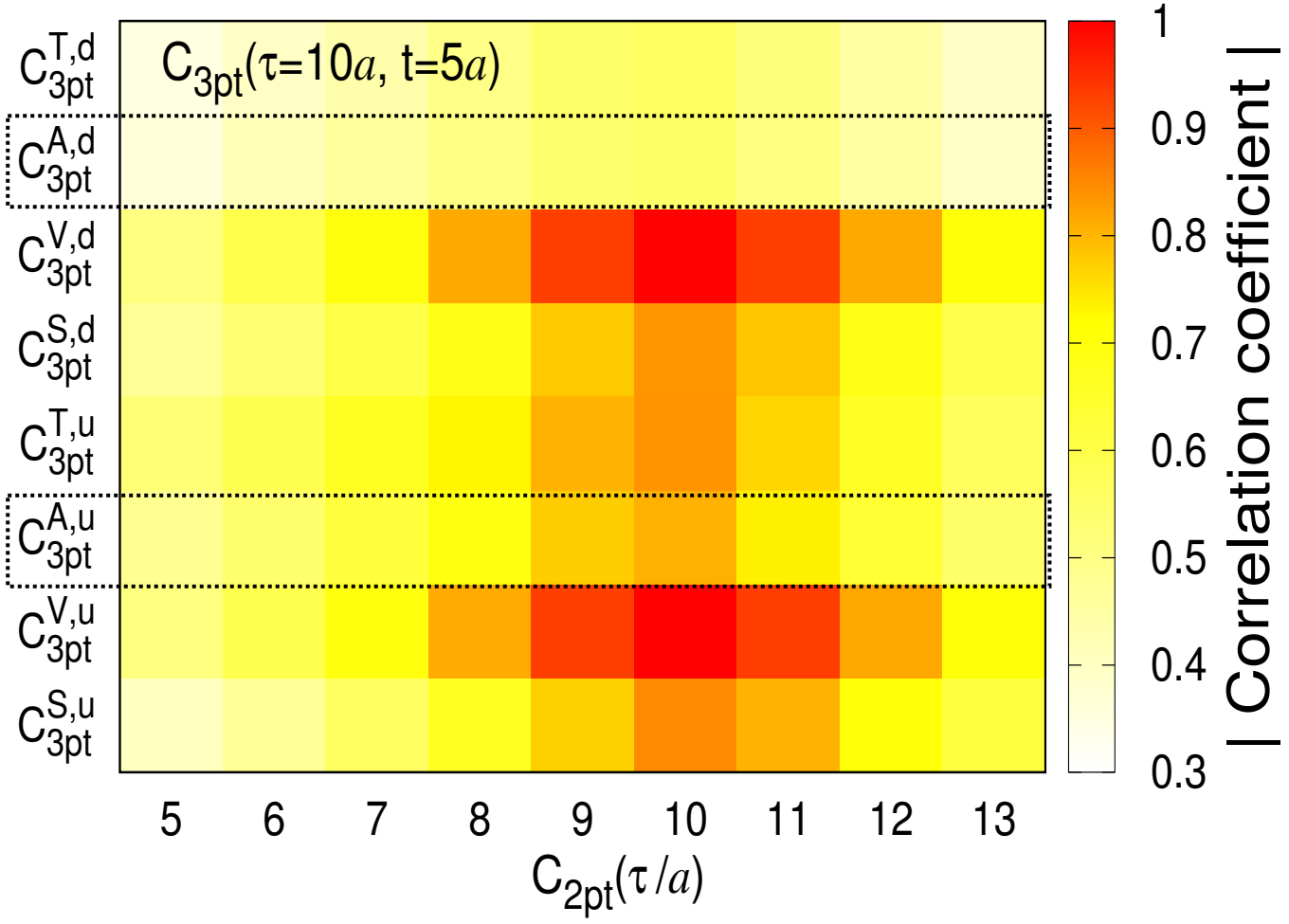
Decision Tree h_0 for $C_{3pt}^{A}(10, 5)$

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

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



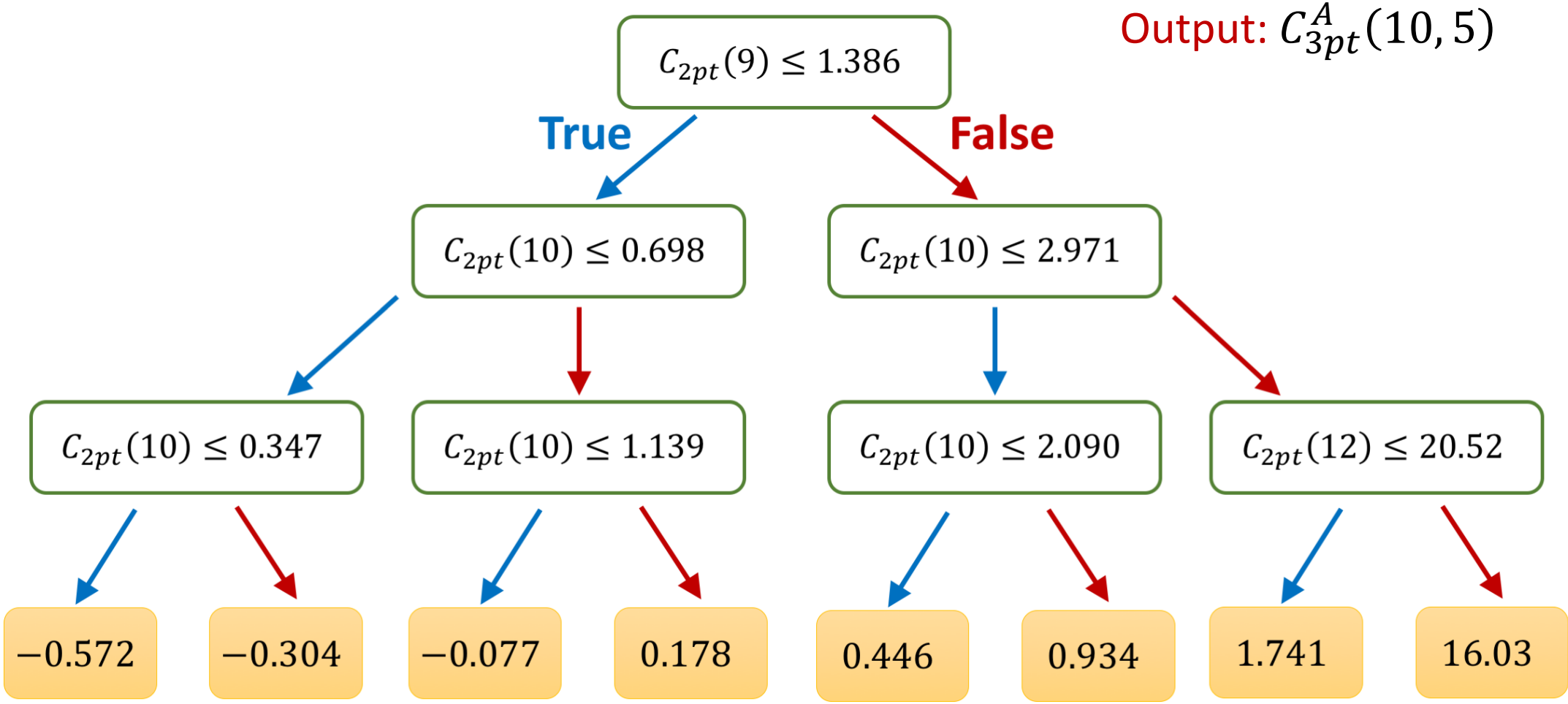
Revisit: Correlation Map of Nucleon Observables



Decision Tree h_5 for $C_{3pt}^{A}(10, 5)$

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

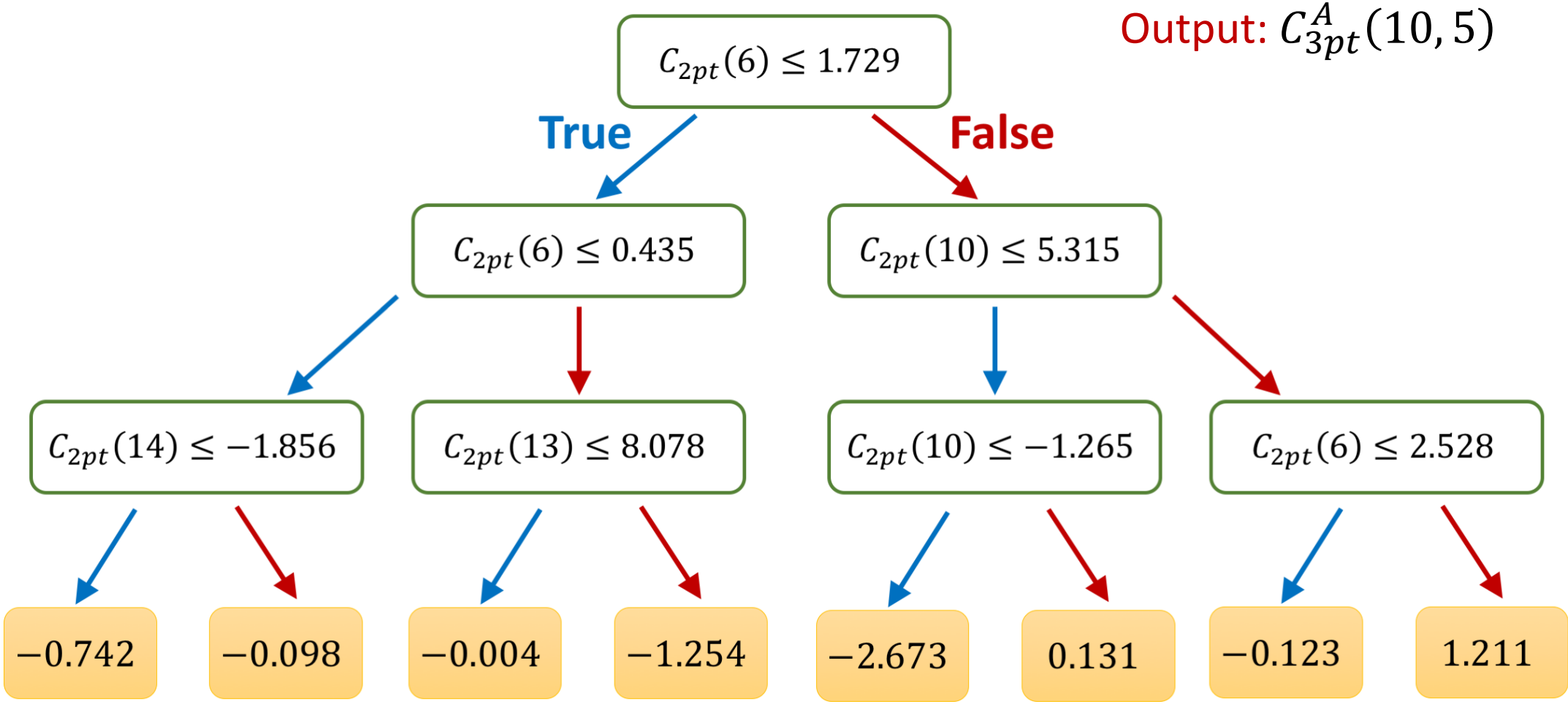
Output: $C_{3pt}^{A}(10, 5)$



Decision Tree h_{30} for $C_{3pt}^{A}(10,5)$

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

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



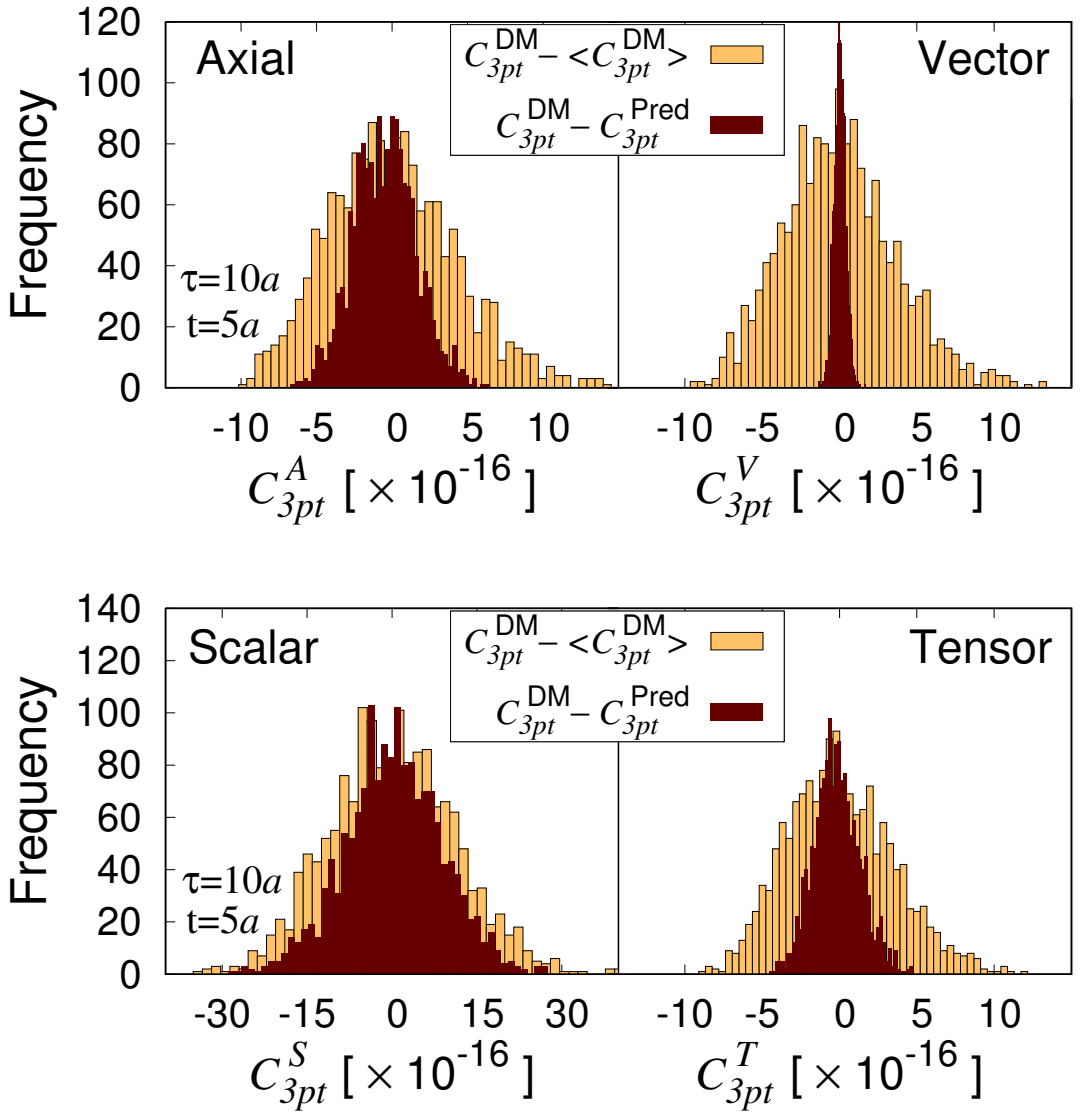
Prediction of C_{3pt} from C_{2pt}

- Training and Test performed for
  - Clover-on-HISQ
  - $a = 0.089$fm, $M_\pi = 313$ MeV
  - Measurements: 2263 confs × 64 srcs
- # of training data: 60 confs
  # of BC data: 620 confs
  # of unlabeled data: 1583 confs
- Prediction error

$$\mathrm{PE} = C_{3pt}^{\mathrm{DirectMeas.}} - C_{3pt}^{\mathrm{Pred}}$$

- Prediction quality for $C_{3pt}^{\Gamma}(10,5)$

$$\frac{\sigma_{PE}}{\sigma_{DM}} = 0.79\ (S), 0.49\ (A), 0.44\ (T), 0.12\ (V)$$



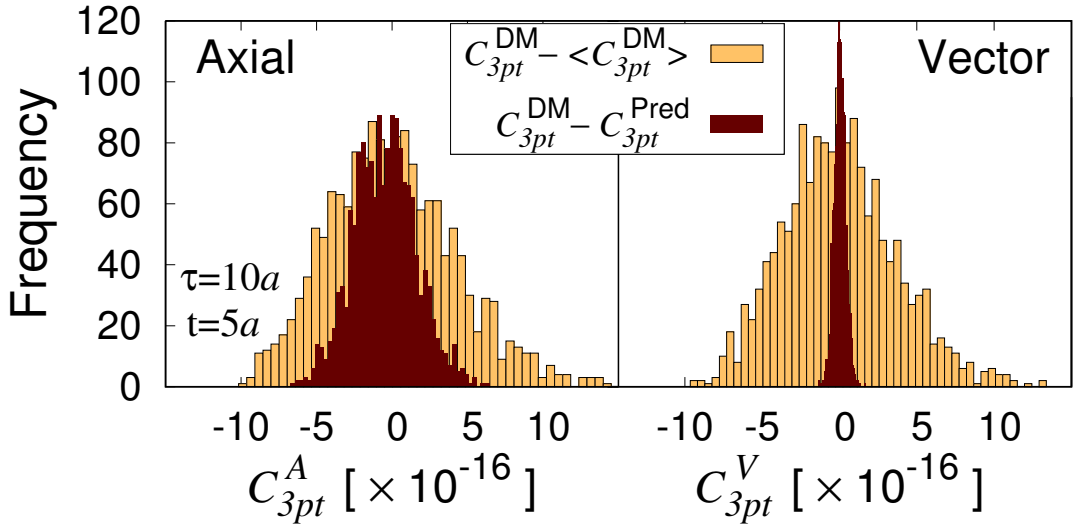
Prediction of C_{3pt} from C_{2pt}

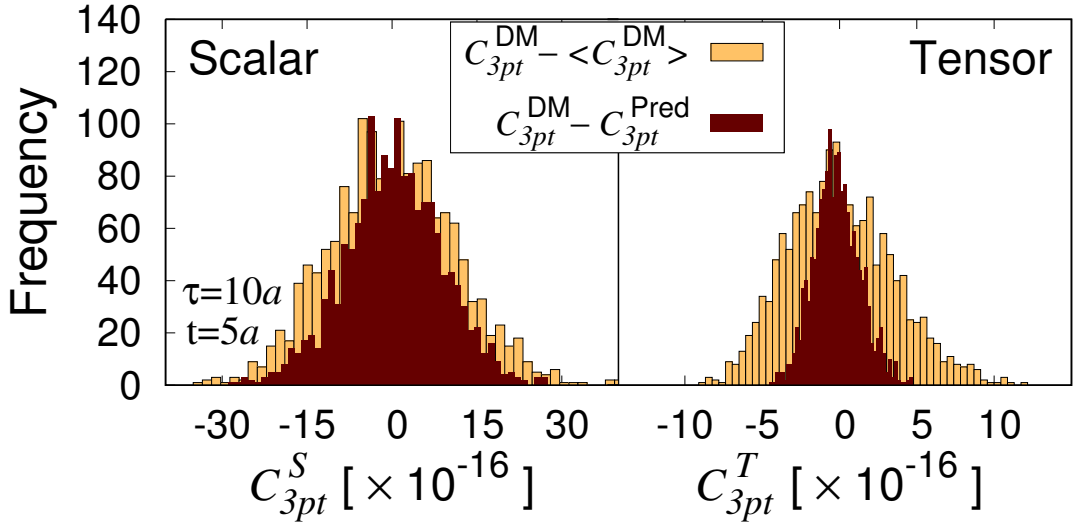
- Prediction quality $C^{\Gamma}_{3pt}(10,5)$

$$\frac{\sigma_{PE}}{\sigma_{DM}} = 0.79\ (S), 0.49\ (A), 0.44\ (T), 0.12\ (V)$$

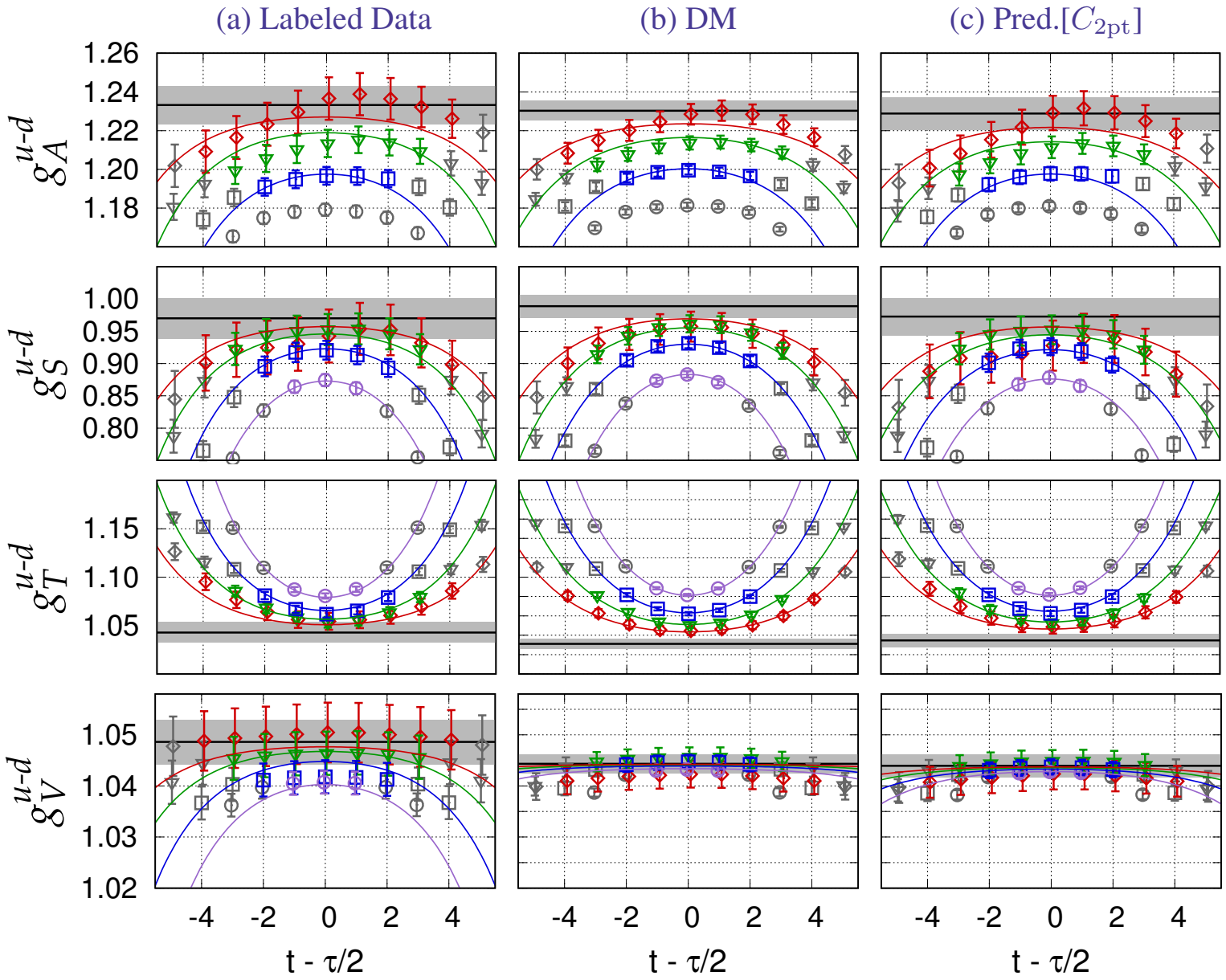
- Predictions of $C^{\Gamma}_{3pt}(10,5)/\langle C_{2pt}(10)\rangle$

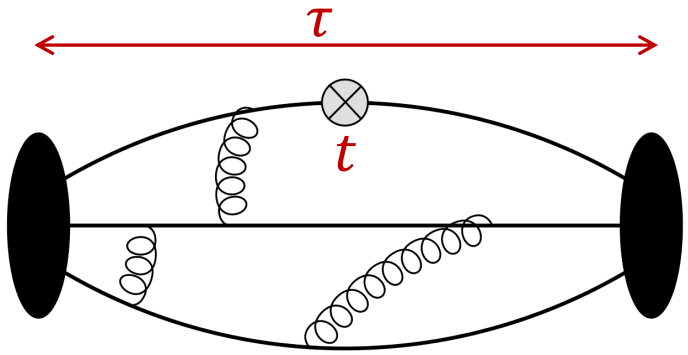
Γ	DM	BC-Prediction	Raw-Prediction	Bias
S	0.936(10)	0.933(15)	0.931(45)	+0.002(46)
A	1.2011(41)	1.1997(48)	1.1999(109)	−0.0003(105)
T	1.0627(34)	1.0638(39)	1.0642(79)	−0.0004(78)
V	1.0462(36)	1.0455(36)	1.0453(39)	+0.0002(20)





Prediction of C_{3pt} from C_{2pt}





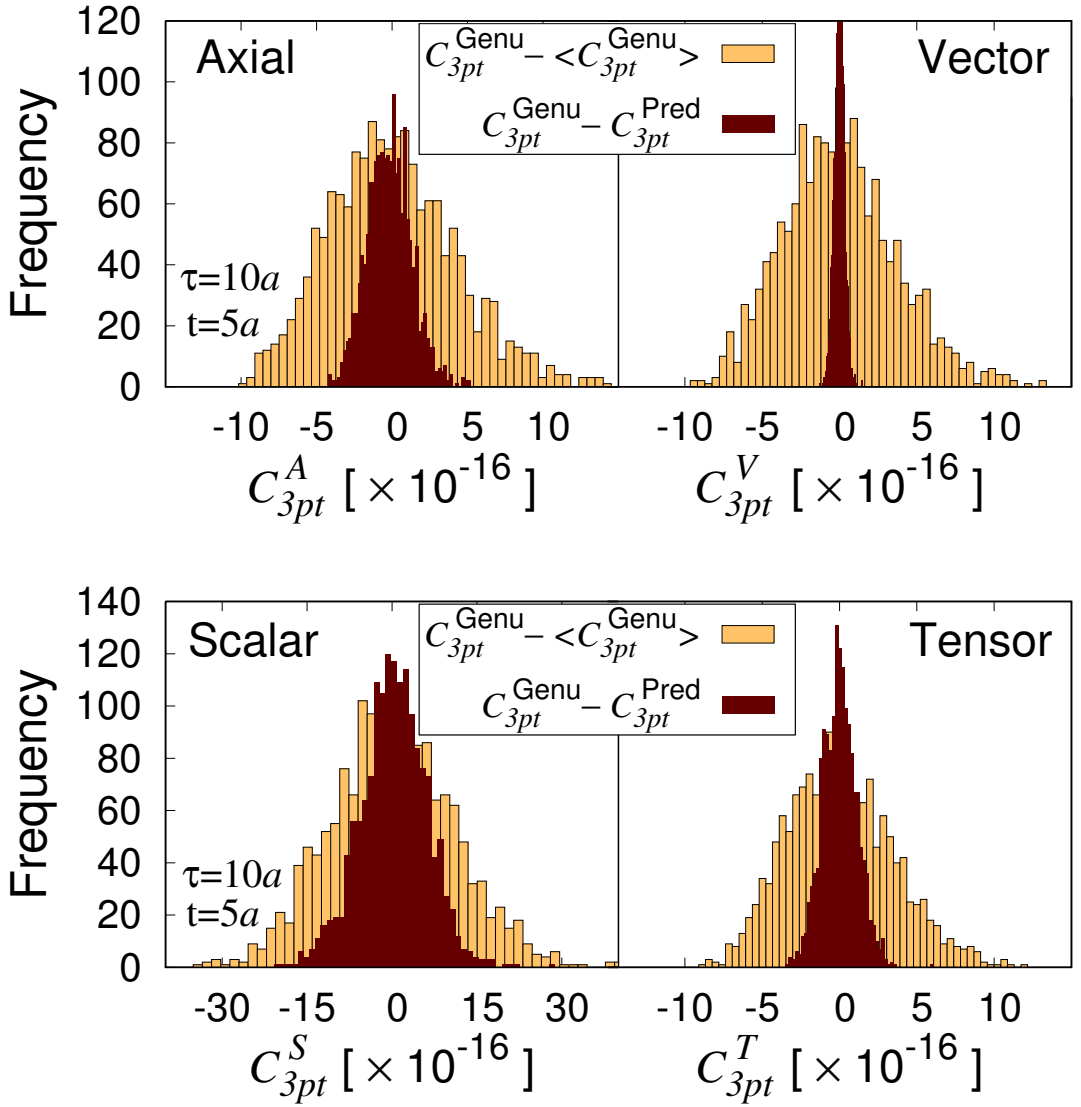
τ=14 τ=12 τ=10 τ=8

Prediction of $\boldsymbol{C_{3pt}(8,10,14)}$ from $\boldsymbol{C_{2pt}}$ and $\boldsymbol{C_{3pt}}(12)$

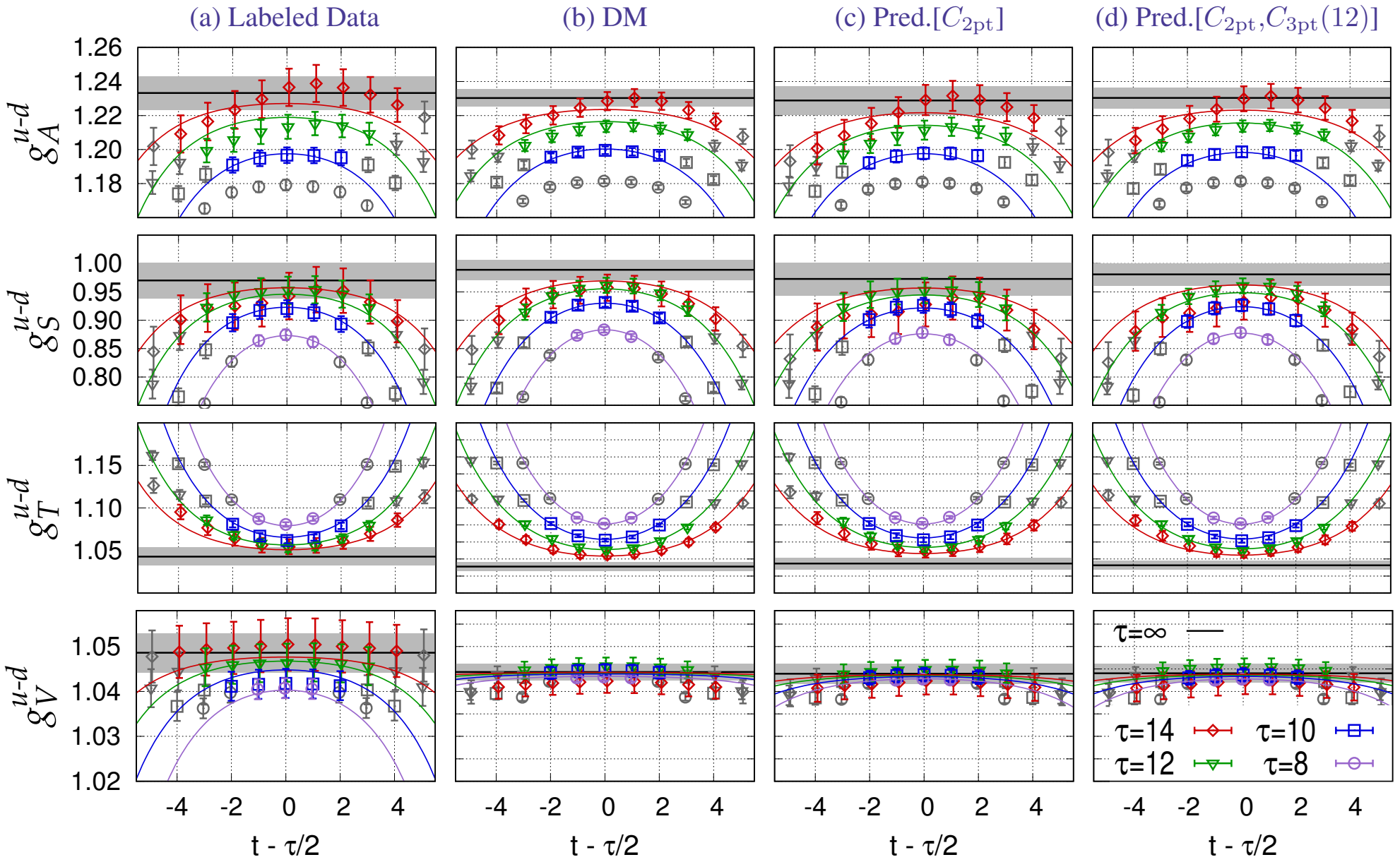
- We need $\boldsymbol{C_{3pt}}$ at four values of source-sink separations ($\tau = 8,10,12,14$) for analysis
- $\boldsymbol{C_{3pt}}(8,10,14)$ can be predicted from $\boldsymbol{C_{2pt}}$ and $\boldsymbol{C_{3pt}}(12)$ data
- Prediction quality $C_{3pt}^{\Gamma}(10,5)$

$$\frac{\sigma_{PE}}{\sigma_{DM}} = 0.53\ (S), 0.35\ (A), 0.33\ (T), 0.10\ (V)$$

(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



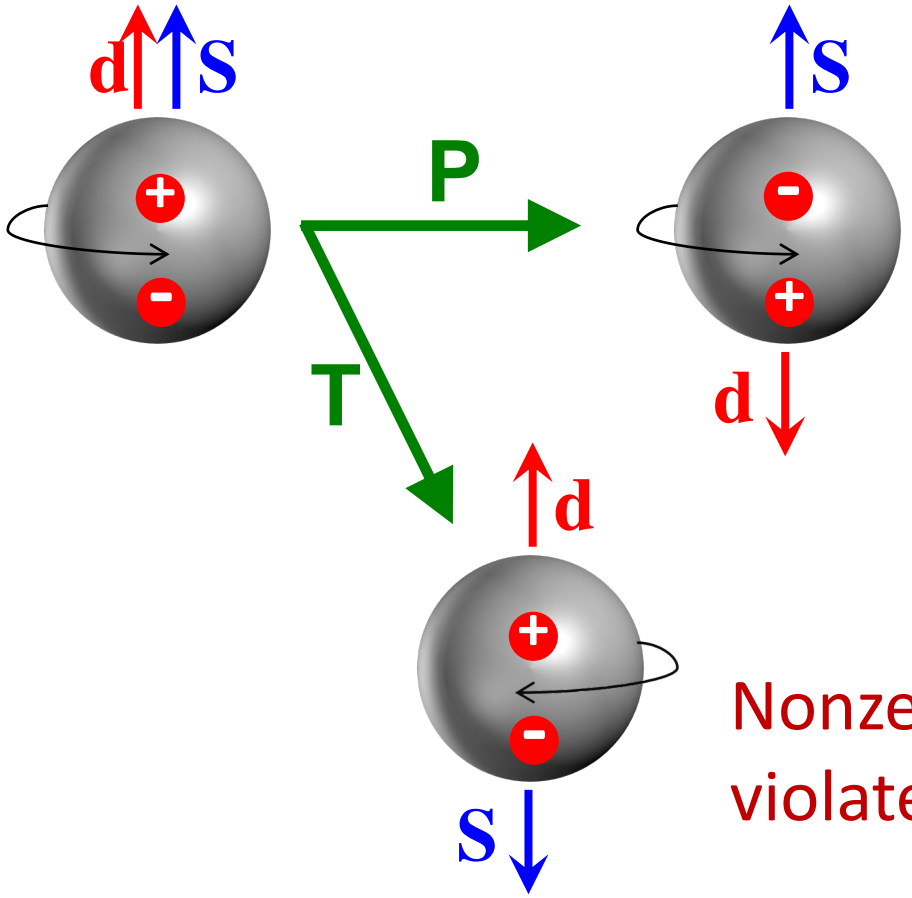
Prediction of C_{3pt} from C_{2pt}

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

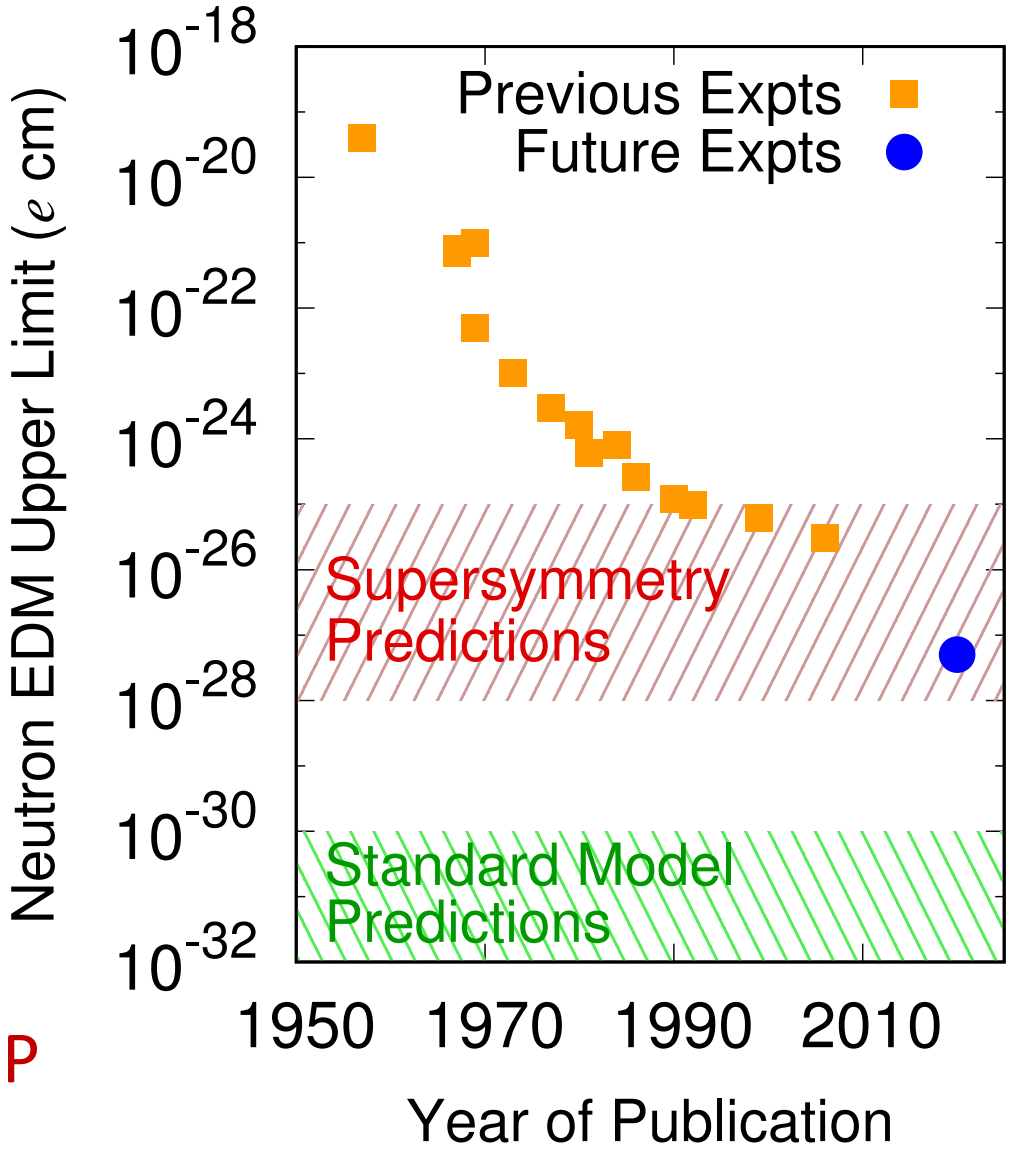
	DM	Pred.$[C_{2\mathrm{pt}}]$	Pred.$[C_{2\mathrm{pt}}, C_{3\mathrm{pt}}(12)]$
g_S	0.989(18)	0.973(29)	0.981(20)
g_A	1.2303(51)	1.2289(83)	1.2304(61)
g_T	1.0311(51)	1.0347(68)	1.0326(54)
g_V	1.0443(19)	1.0439(22)	1.0440(21)
	2263 DM (Direct Meas.)	680 DM + 1583 Pred.	680 DM + 1583 Pred.

Neutron EDM and CP Violation

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



Nonzero nEDM violates P and T, so CP



Effective CPV Lagrangian

$$\mathcal{L}_{\mathrm{CPV}}^{d\le 6} = -\frac{g_s^2}{32\pi^2}\bar{\theta} G\tilde{G}$$ 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

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

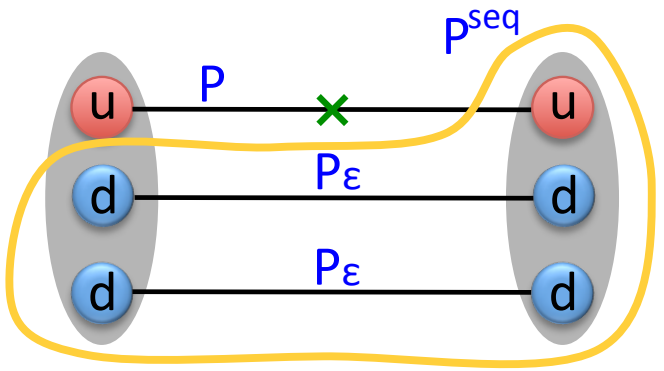
$$S_{cEDM} = -\frac{i}{2}\int d^4x \;\; \tilde{d}_q g_s \bar{q}(\sigma \cdot G)\gamma_5 q$$

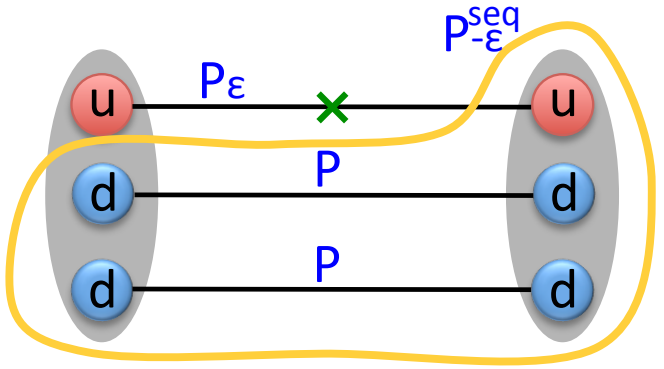
- Schwinger source method

  Include cEDM term in valence quark propagators by modifying Dirac operator

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

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





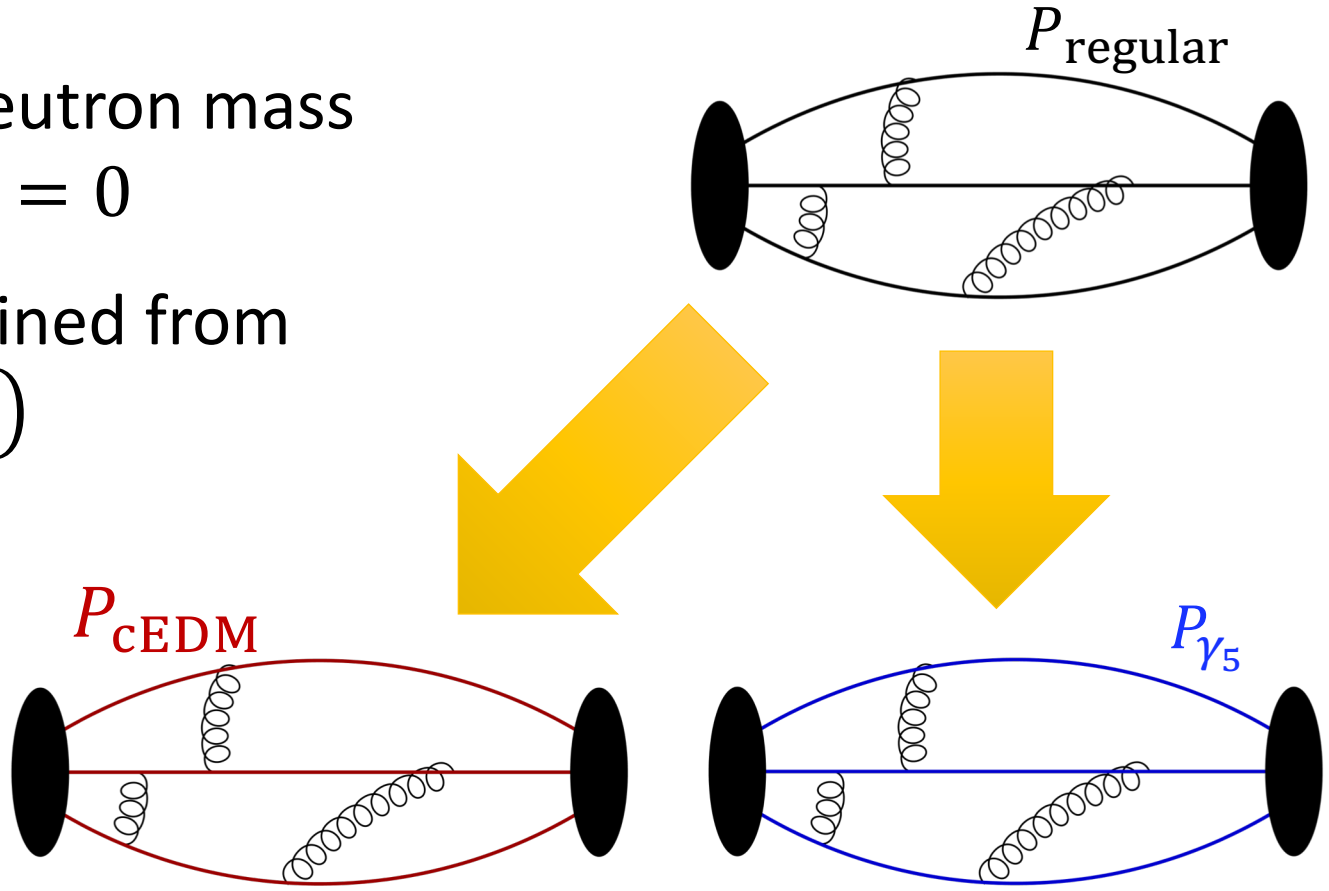
Prediction of C_{2pt}^{CPV} from C_{2pt}

- Predict C_{2pt} for cEDM and γ_5 insertions from C_{2pt} without CPV
- CPV interactions ➔ phase in neutron mass

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

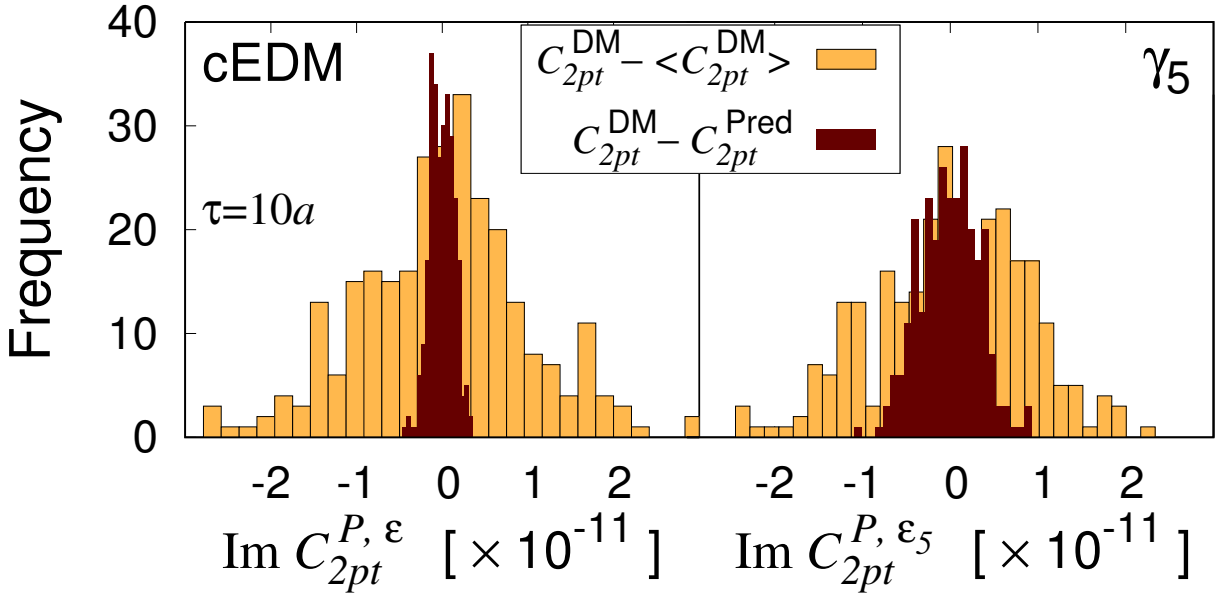
- At leading order, α can be obtained from

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



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



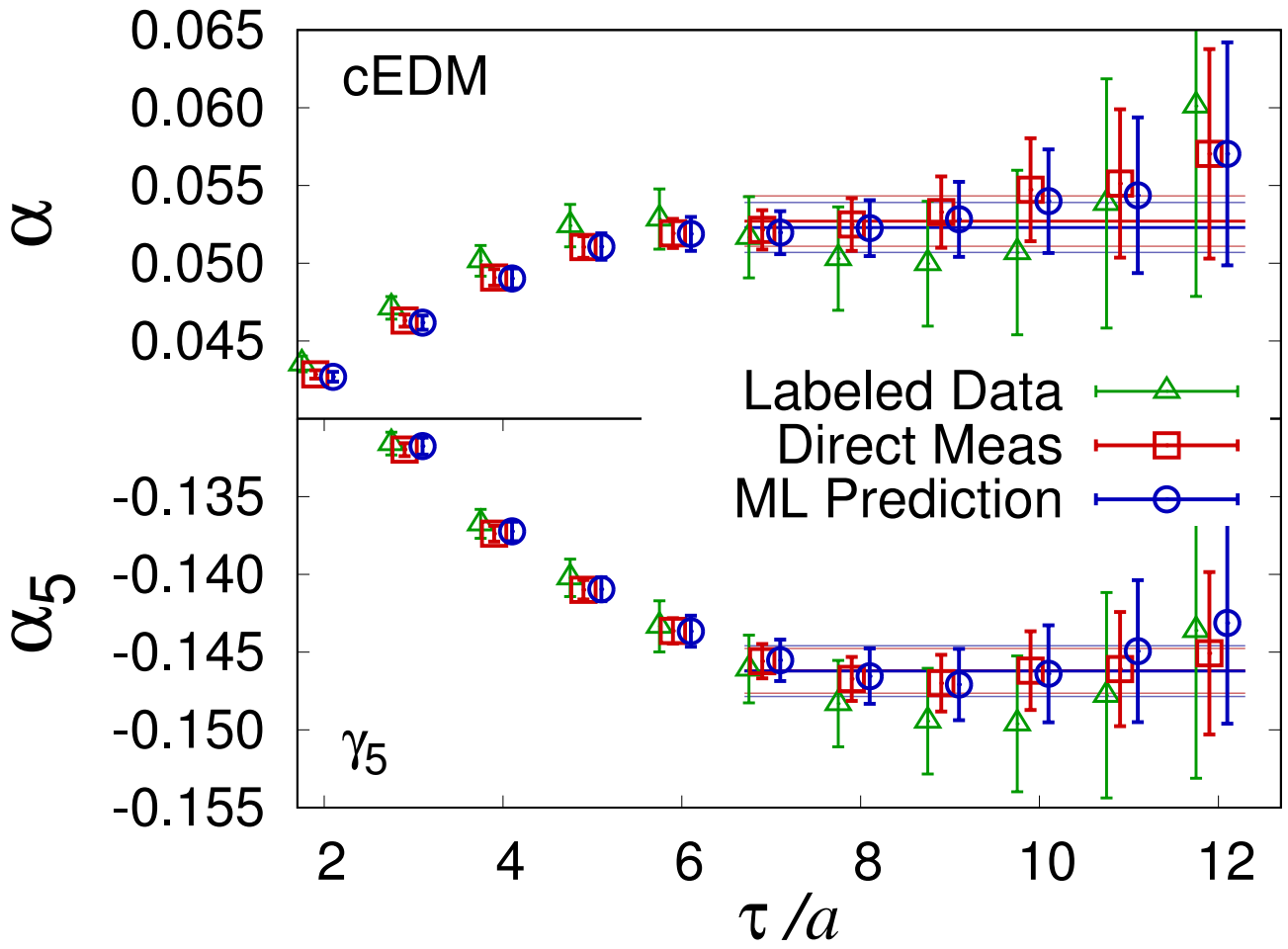
Input:

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

Boosted Decision Tree Regression

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

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



- α $(\mathbf{cEDM})$

  DM: 0.0527(17)

  Prediction: 0.0525(18)

- α_5 (γ_5)

  DM: -0.1463(14)

  Prediction: -0.1460(17)

➢ DM: DM on 400 confs

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

Comparison of Regression Models

	Linear Regression	BDT	Neural Network
Speed	Fastest	Fast	Slow
Performance	Bad for nonlinear	Okay	Possibly better
Tuning Parameters	None or a few	Few; not sensitive	Many; sensitive
Overfitting Risk	Very Low	Low	High
Training Data Requirement	Small	Medium	Large
Interpretability	Yes	Somewhat	Not likely

BDT with scikit-learn Python ML Library

```
>>> 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]])     # 3²+2×4 = 17
array([15.20630936])

>>> gbr.predict([[6,3]])     # 6²+2×3 = 42
array([42.77231812])

>>> gbr.predict([[8,5]])     # 8²+2×5 = 74
array([74.14274825])
```

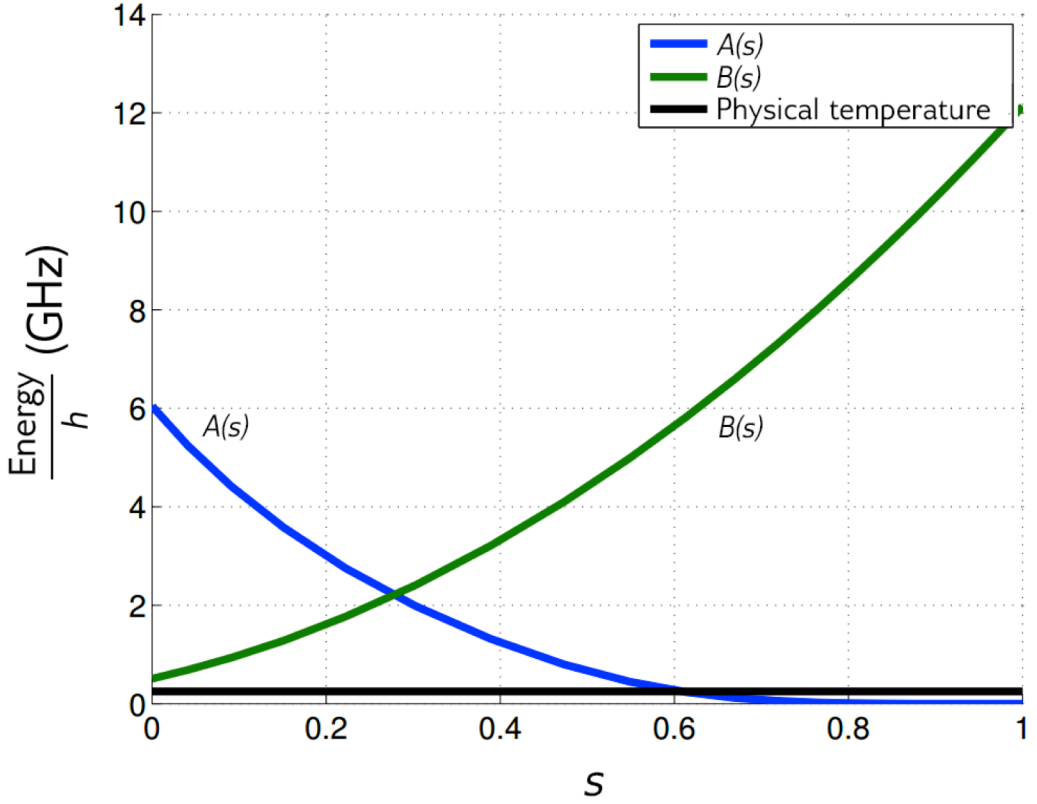
$$X = [[a_1, b_1], [a_2, b_2], \dots]$$
$$y = [a_1^2 + 2b_1, a_2^2 + 2b_2, \dots]$$

Machine Learning Regression on D-Wave Quantum Annealer

D-Wave Quantum Computer

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

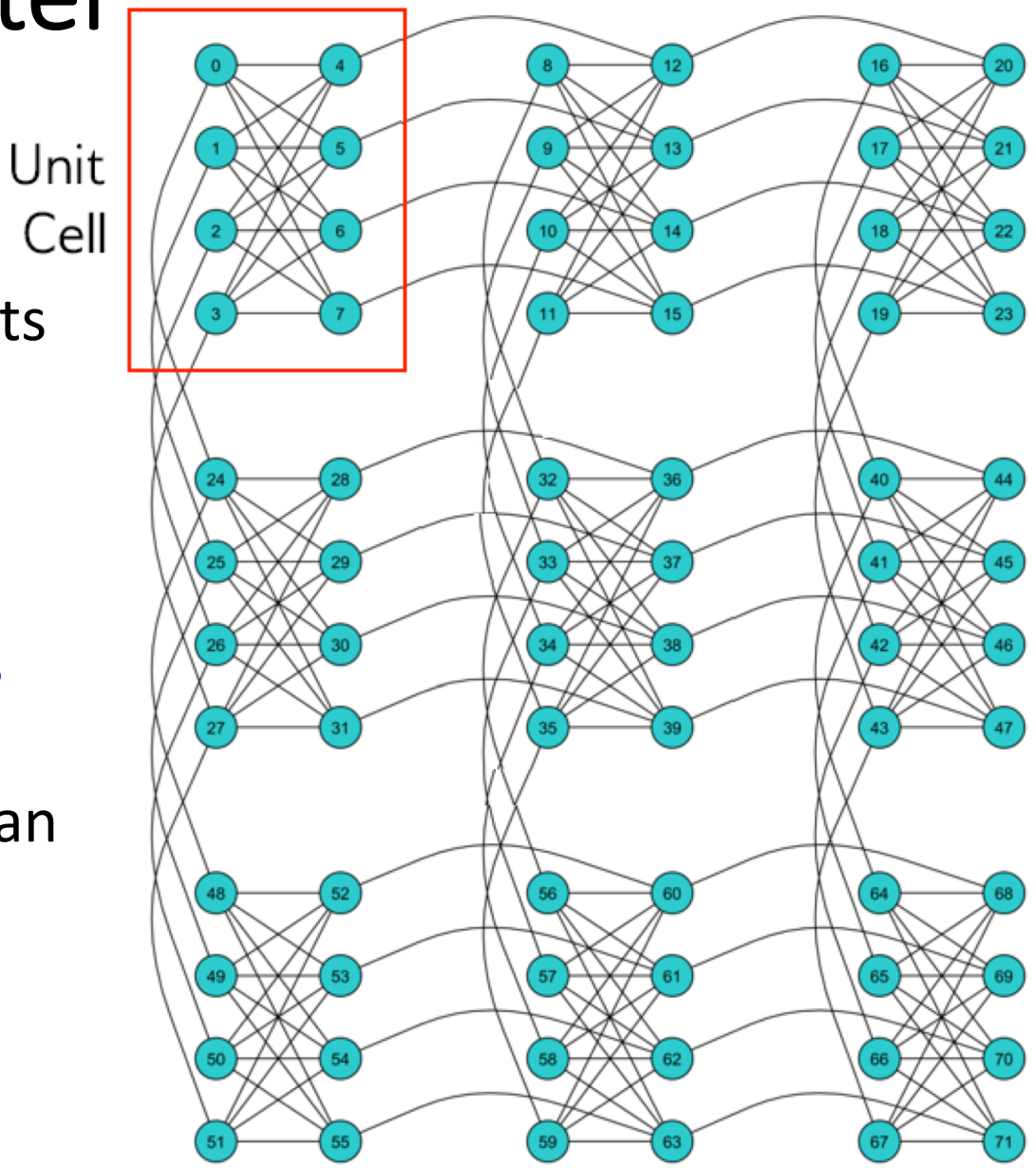
- $\hat{\sigma}_{x,z}^{(i)}$: Pauli matrices operating on a qubit q_i
- h_i, $J_{i,j}$: qubit biases and coupling strengths; set by user
- Qubits collapse to classical states (0 or 1), the lowest-energy state of the final H



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

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

- Unsupervised ML algorithm
- Find dictionary $\mathbf{\Phi}$ and sparse representation $\vec{a}^{(k)}$ from which input data can be reconstructed by

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

- The representation is sparse because the λ-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

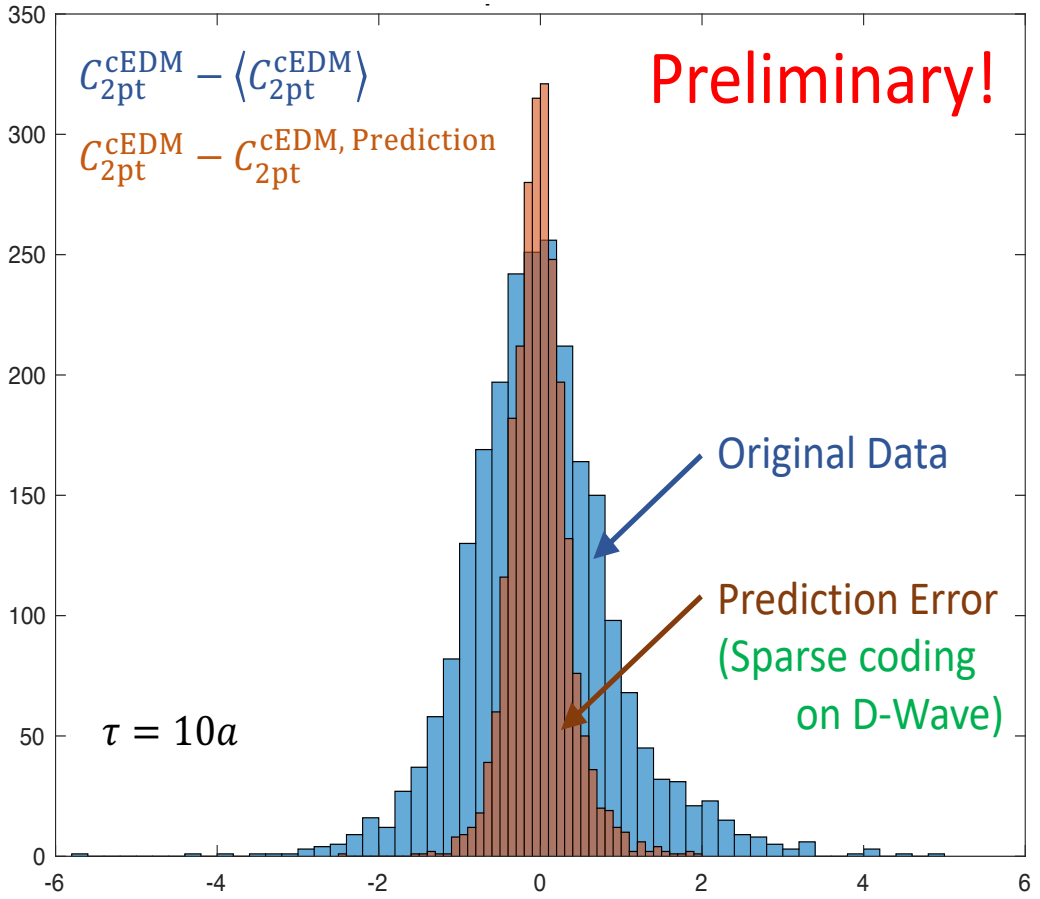
$$H(\vec{h}, \boldsymbol{Q}, \vec{a}) = \sum_i a_i h_i + \sum_{\langle i,j \rangle} Q_{ij} a_i a_j$$

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

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



Distribution of CPV correlators and ML prediction error

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.