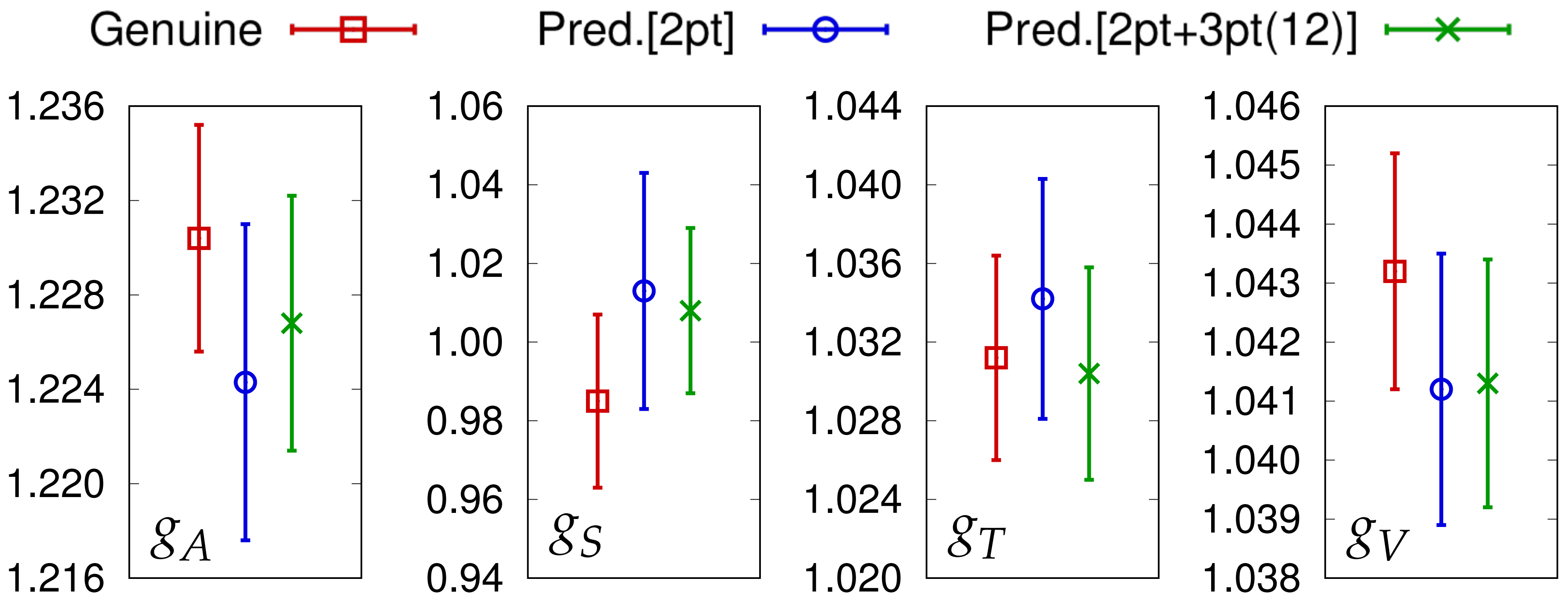
Do NOT measure correlated observables, but train an Artificial Intelligence to predict them

Boram Yoon
Los Alamos National Laboratory

Lattice 2018, East Lansing, Michigan, USA, July 22-28, 2018

arXiv:1807.05971

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



Systematic error due to ML prediction included in errorbars

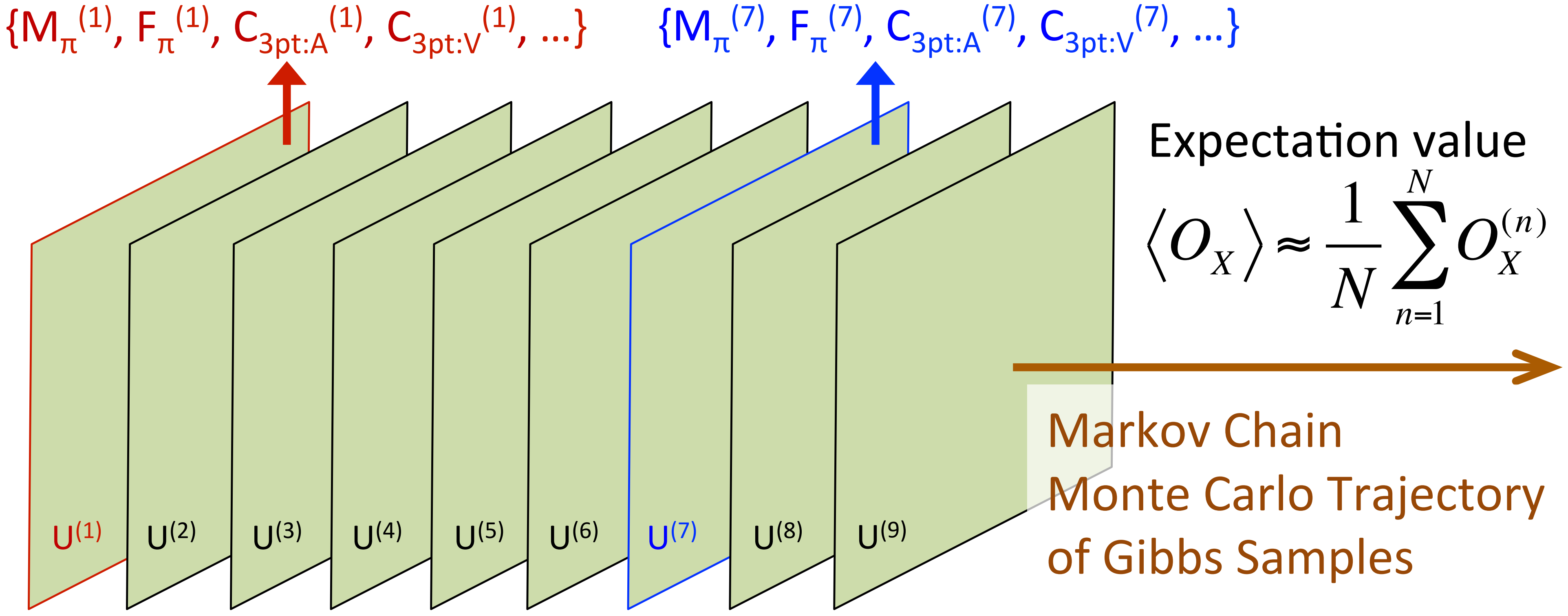
- **Genuine**

  Directly measured on 2263 confs

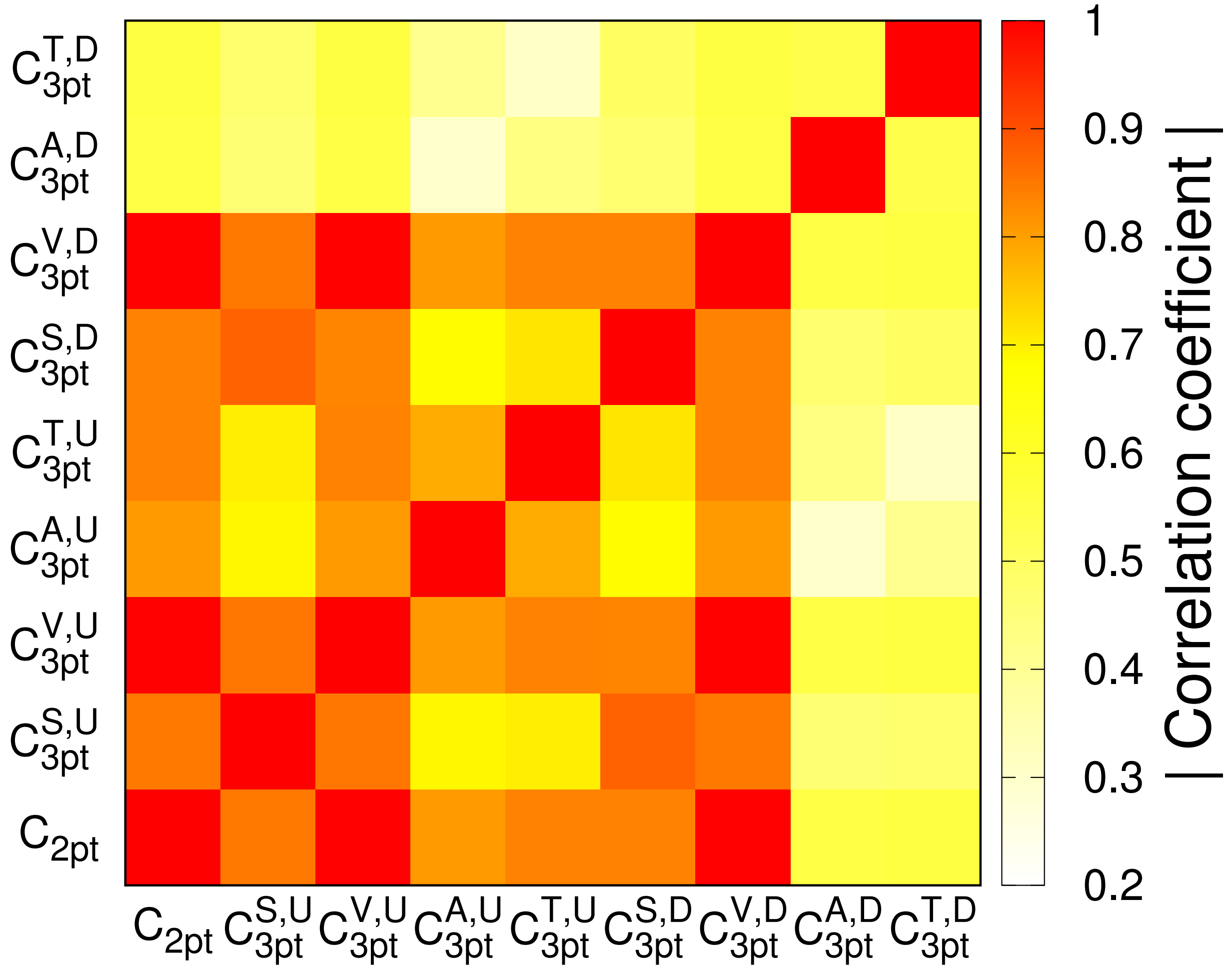
- **ML Prediction**

  Directly measured on 400 confs
  + ML prediction on 1863 confs

Lattice QCD Observables are Correlated



Correlation Map of Nucleon Observables



- Correlation between proton(uud) 3-pt and 2-pt correlation functions
- Clover-on HISQ
  $a = 0.089$ fm, $M_\pi = 313\text{MeV}$
  $\tau = 10a,\ t = \tau/2$
- Using these correlations, C_{3pt} can be estimated from C_{2pt} on each configuration

Machine Learning

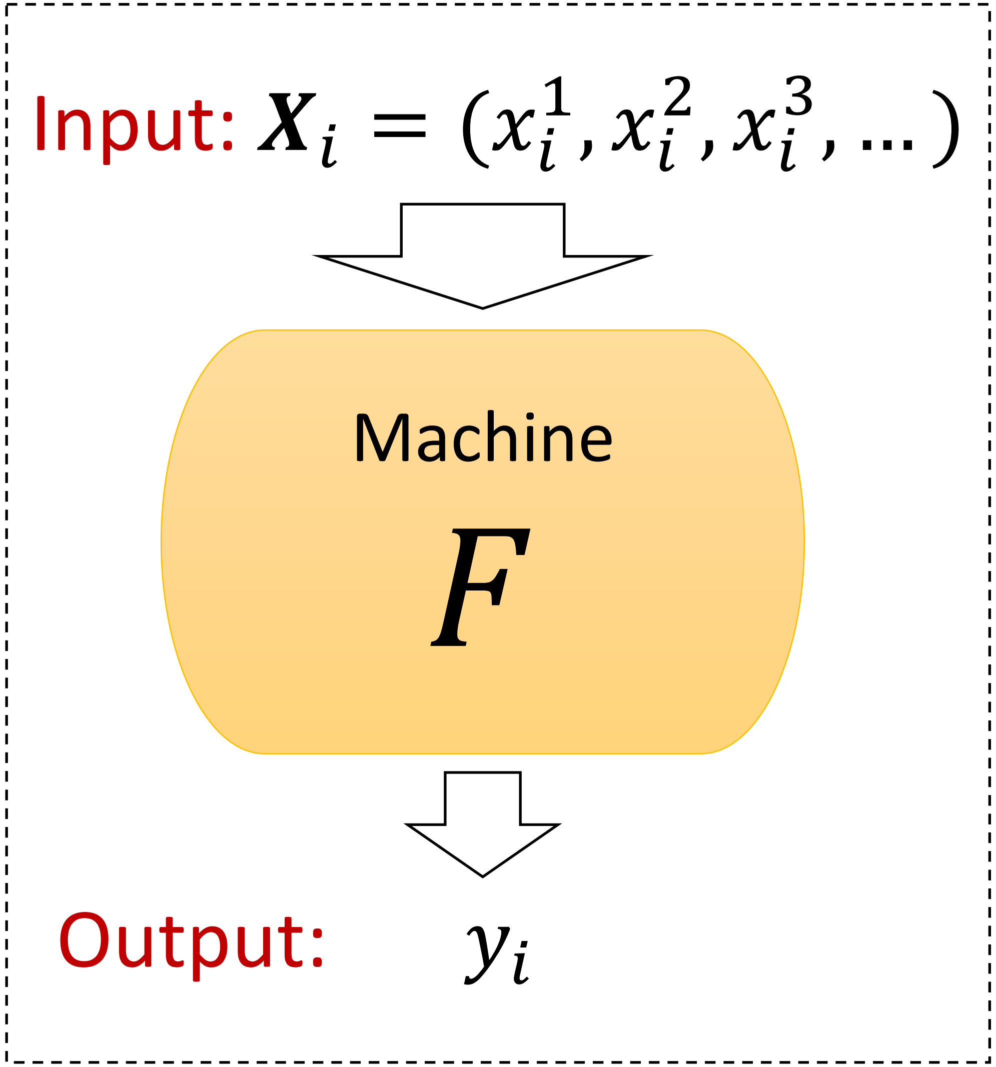
- One can consider the machine learning (ML) process as a data fitting
- The machine F has very general fitting functional form with huge number of free parameters
- The free parameters are determined from large number of training data:

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

- For example,

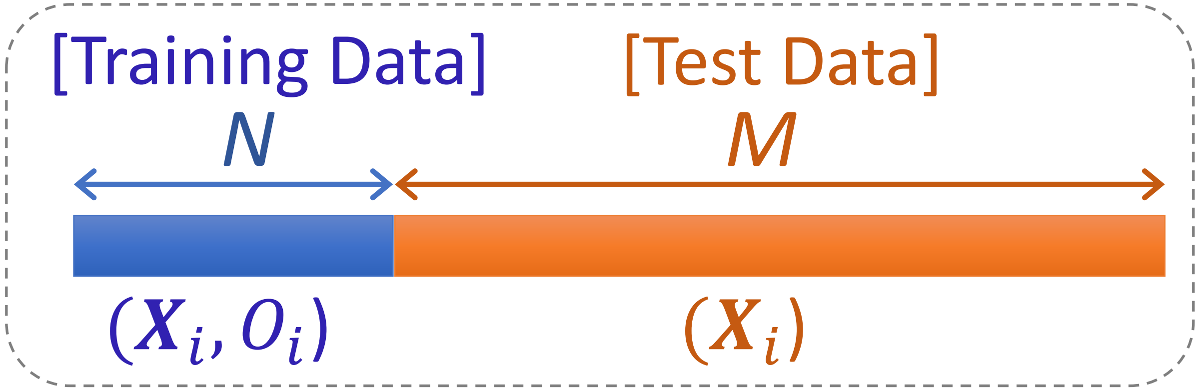
  $\boldsymbol{X}_i$: pixels of a picture

  y_i : "***cat***" or "***dog***"



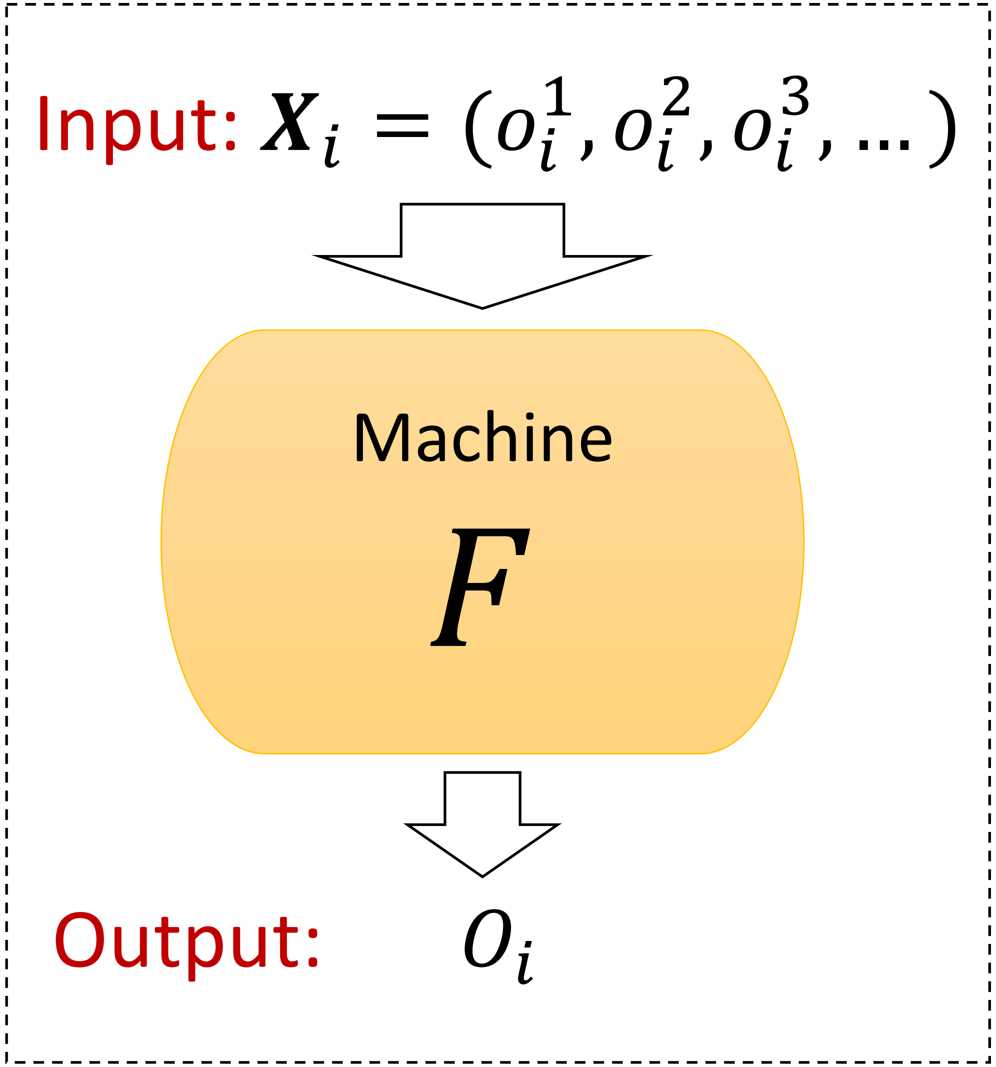
Machine Learning on Lattice QCD Observables

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



1) **Train** machine ***F*** to yield O_i from $\boldsymbol{X}_i$ on the Training Data
2) **Predict** O_i of the Test data from $\boldsymbol{X}_i$

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



Prediction Bias

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

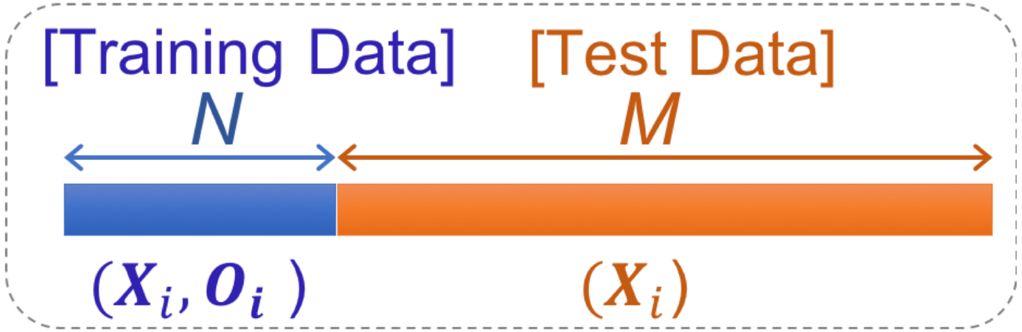
$$\overline{O} = \frac{1}{M}\sum_{i=N+1}^{N+M} O_i^P$$

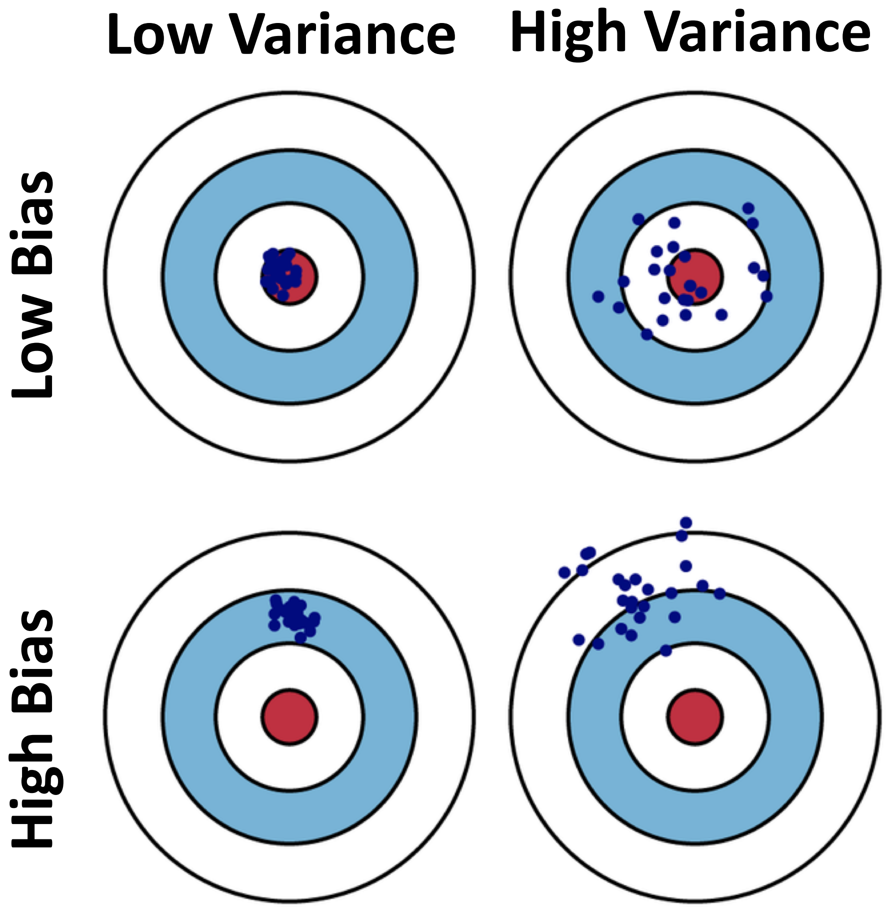
is not correct due to prediction bias

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

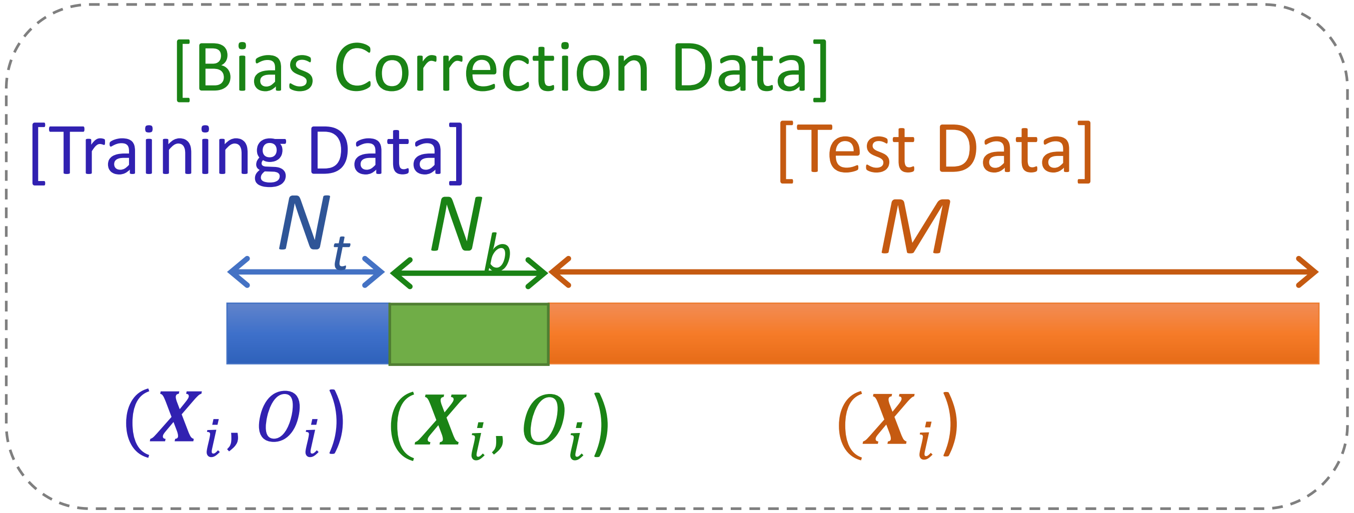
$$\langle O_i^P \rangle \neq \langle O_i \rangle$$

Bias $= \langle O_i^P \rangle - \langle O_i \rangle$





Bias Correction

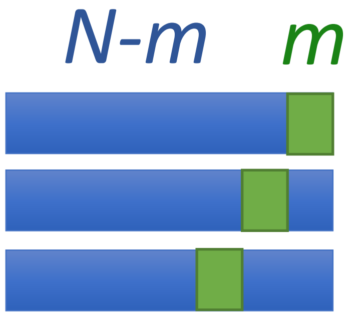


- Average of predictions on test data with bias correction

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

- Expectation value, $\langle\overline{O}\rangle = \langle O_i^P\rangle + \langle O_i - O_i^P\rangle = \langle O_i\rangle$
- Training data should not overlap with bias correction data
- Not efficient: small training/bias correction data

Bias Correction – Cross Validation



$l = 1 \quad \rightarrow F^1,\ O_i^{P,1} = F^1(\boldsymbol{X}_i)$

$l = 2 \quad \rightarrow F^2,\ O_i^{P,2} = F^2(\boldsymbol{X}_i)$

$l = 3 \quad \rightarrow F^3,\ O_i^{P,3} = F^3(\boldsymbol{X}_i)$

… … …

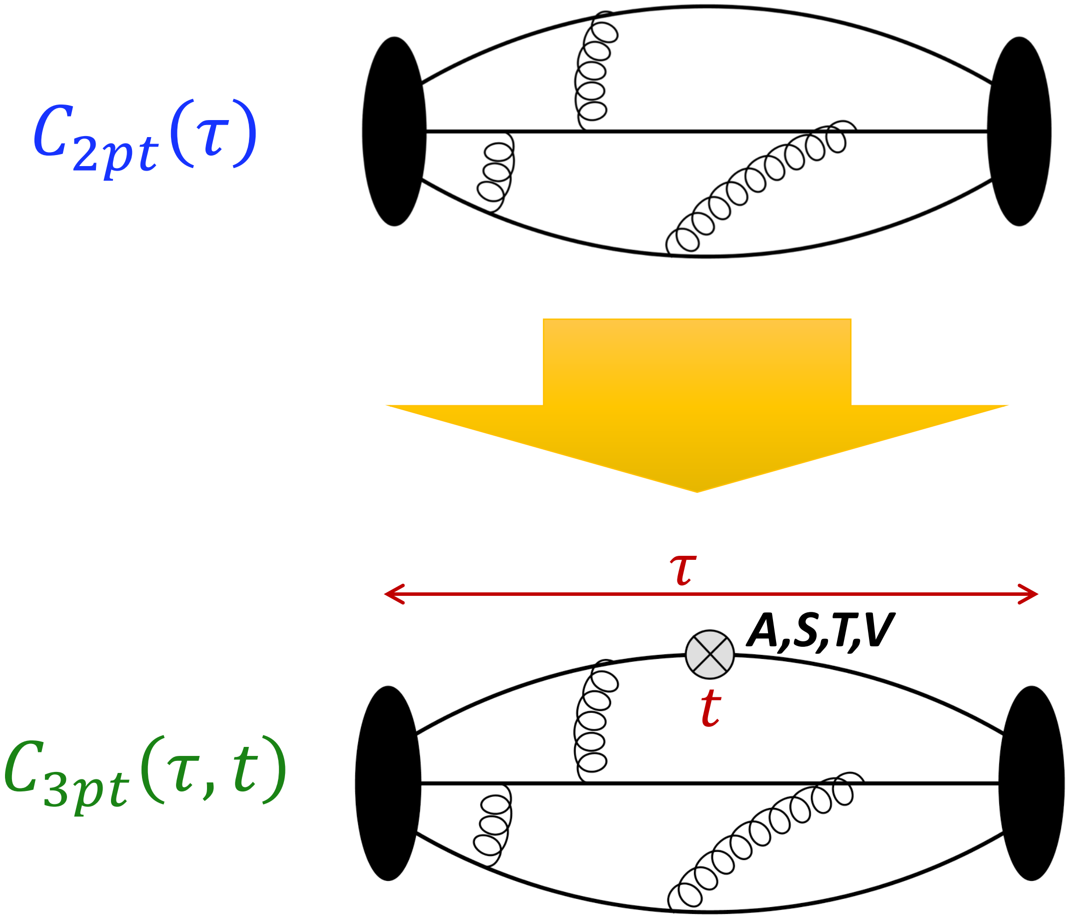
- Average of predictions on test data with bias correction

$$\overline{O} = \frac{1}{L}\sum_{l=1}^{L}\left(\frac{1}{M}\sum_{i=N+1}^{N+M} O_i^{P,l} + \frac{1}{m}\sum_{k=1}^{m}\left(O_k^l - O_k^{P,l}\right)\right)$$

$L = N/m, m \ll N$

- Full training data & precise bias estimation
- Systematic error of ML prediction naturally included in error estimation

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



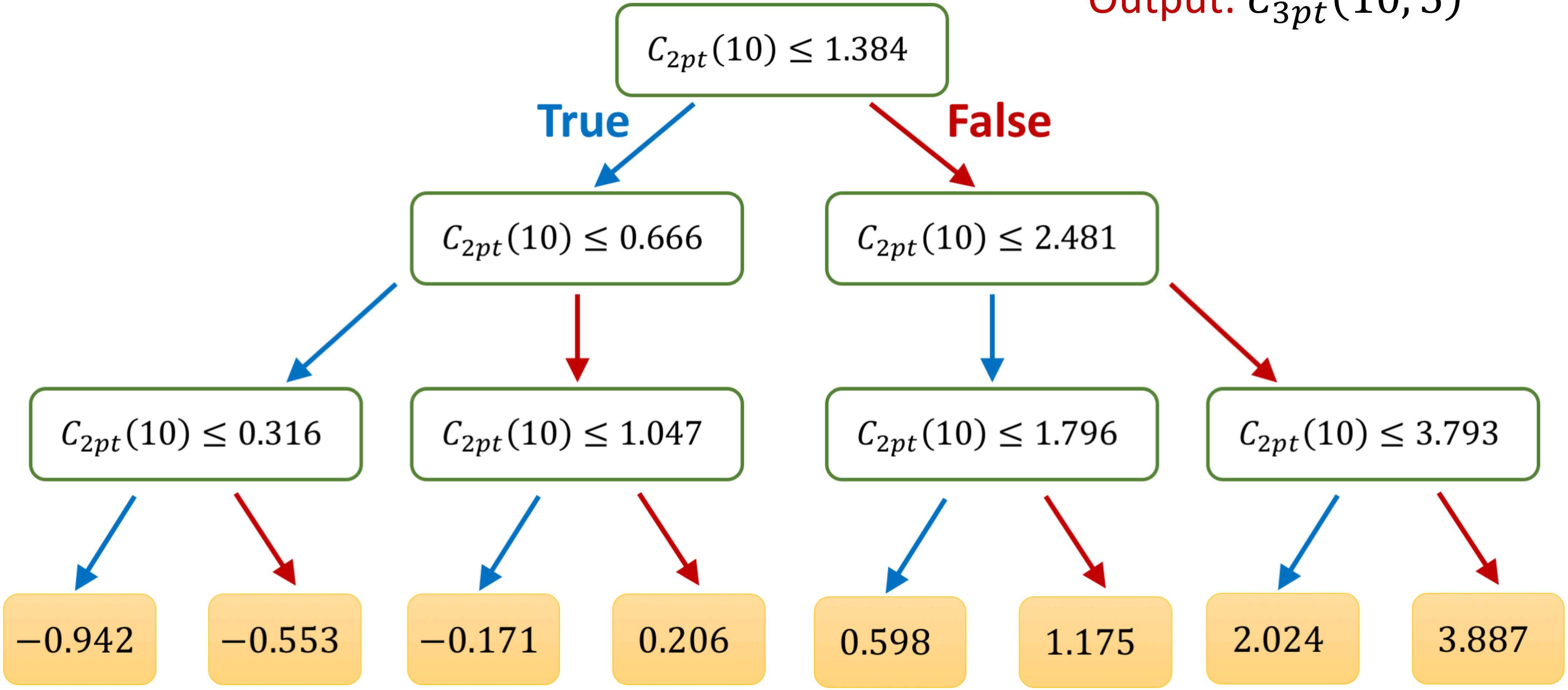
Input: $\boldsymbol{X}_i = \{C_{2pt}(0 \le \tau/a \le T_{\mathrm{max}})\}$

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



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

Boosted Decision Tree (BDT)

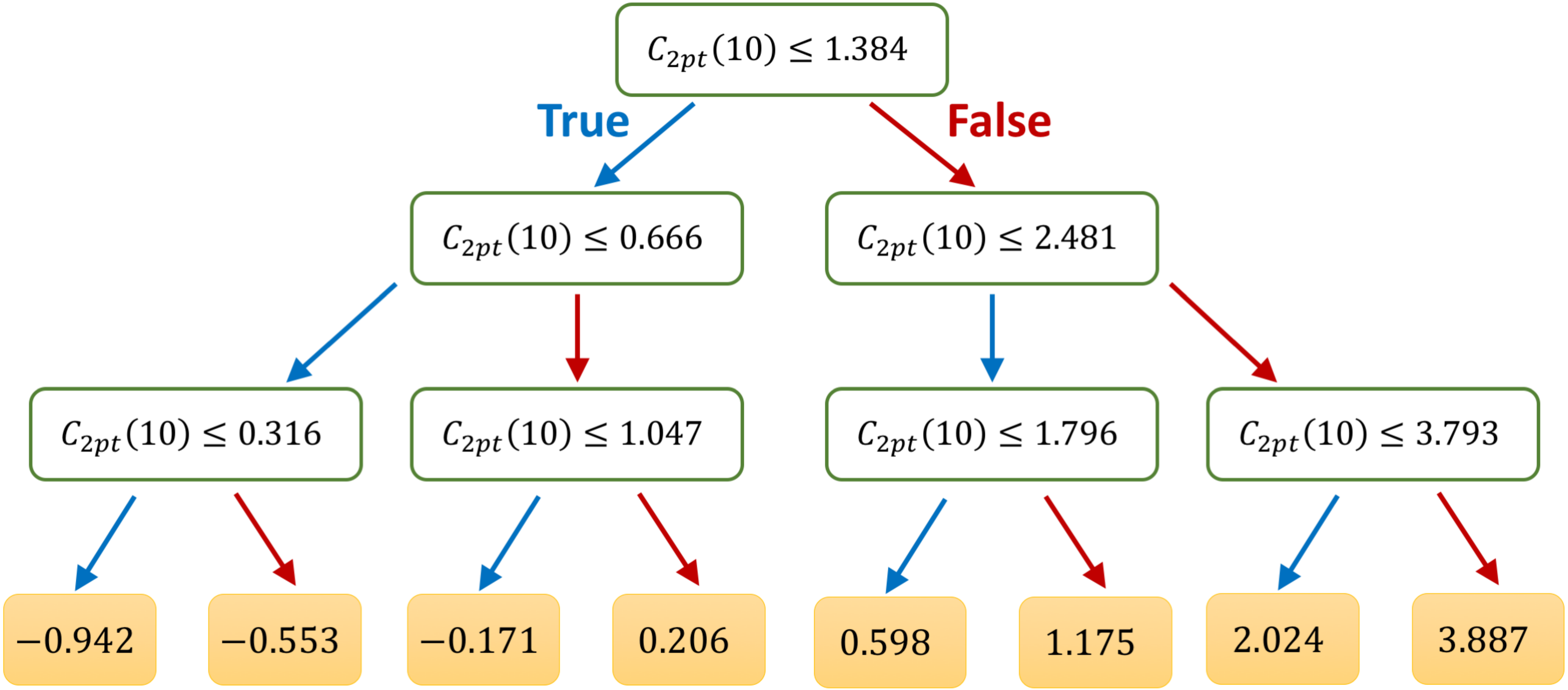
- **Iterative boosting**

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]
…
$F_n = F_{n-1} + h_n$

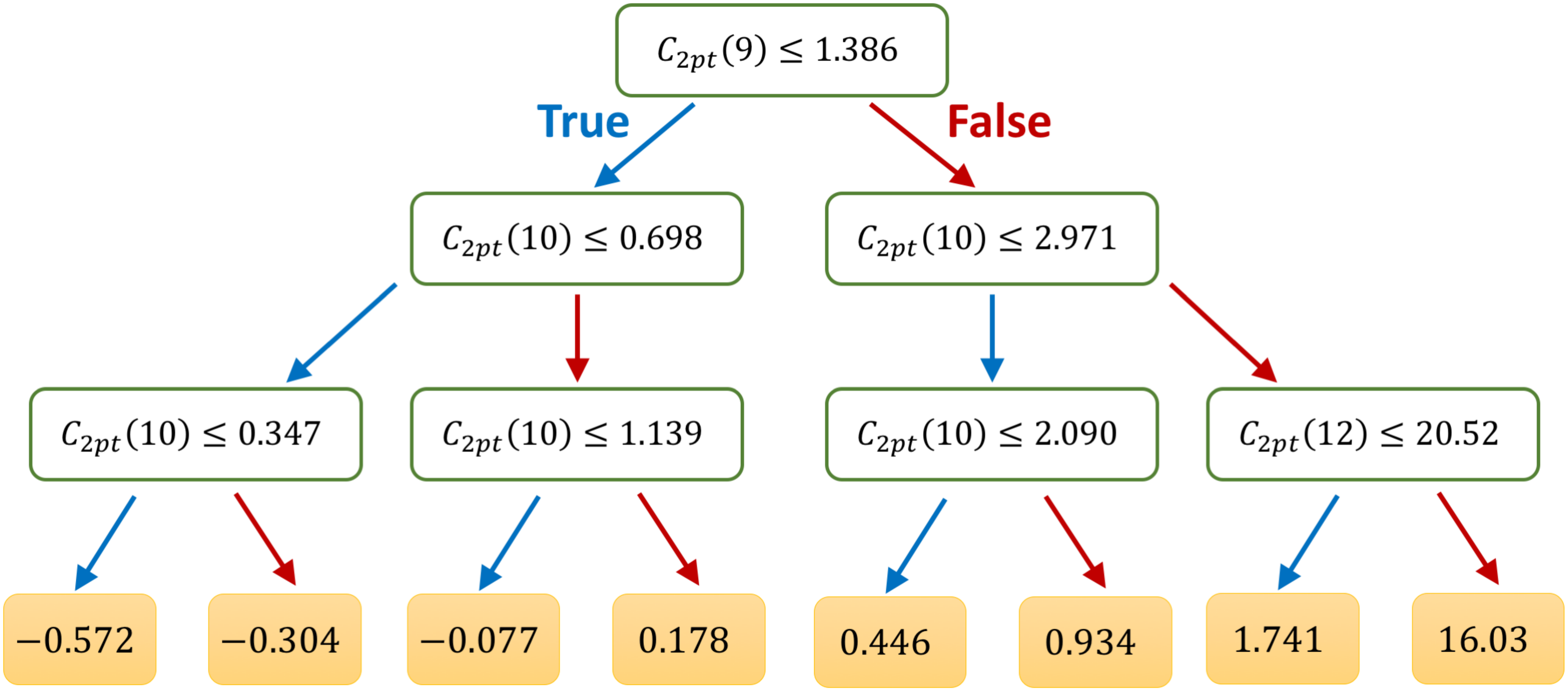
$$F(\boldsymbol{X}) = F_{N_{boost}}(\boldsymbol{X})$$

- In this study, $N_{boost} = 200 - 500$

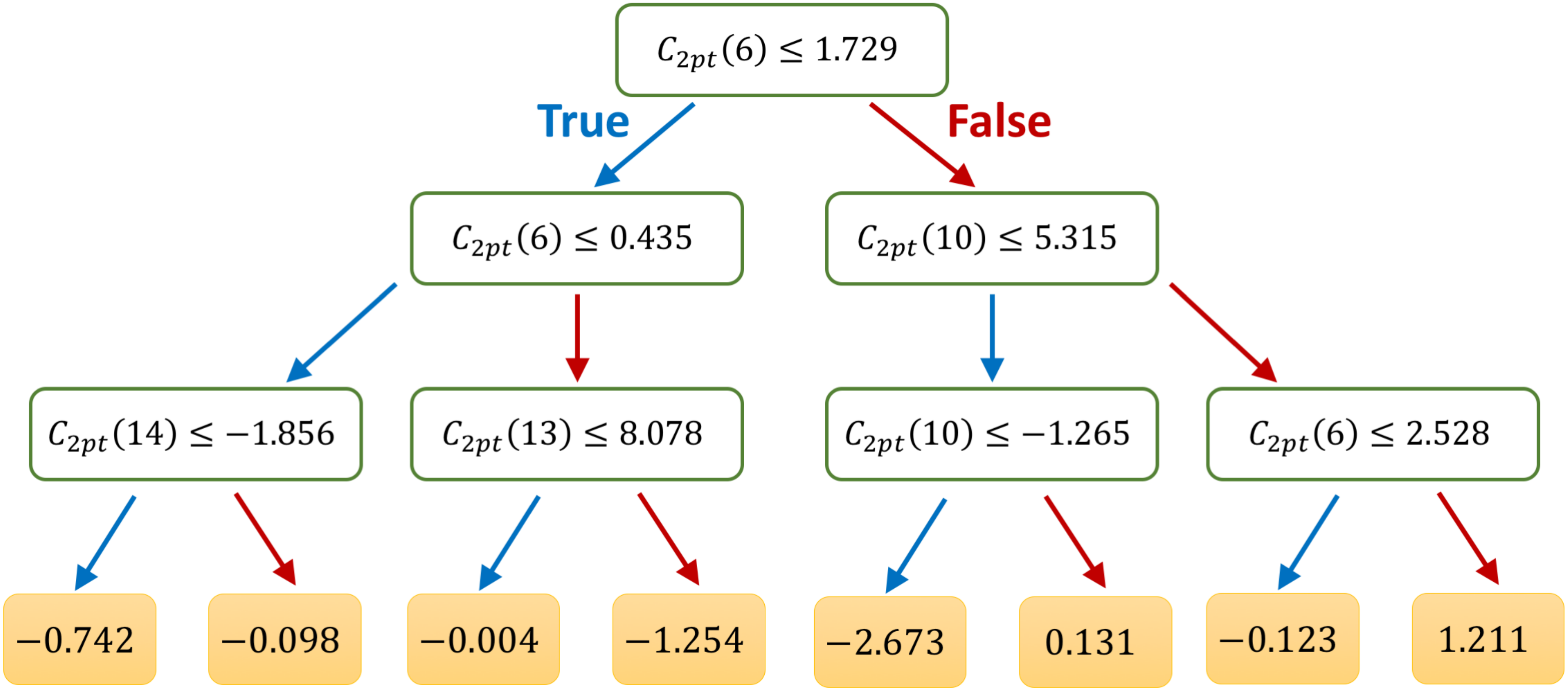
Decision Tree h_0 for $C_{3pt}^{A}(10,5)$



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



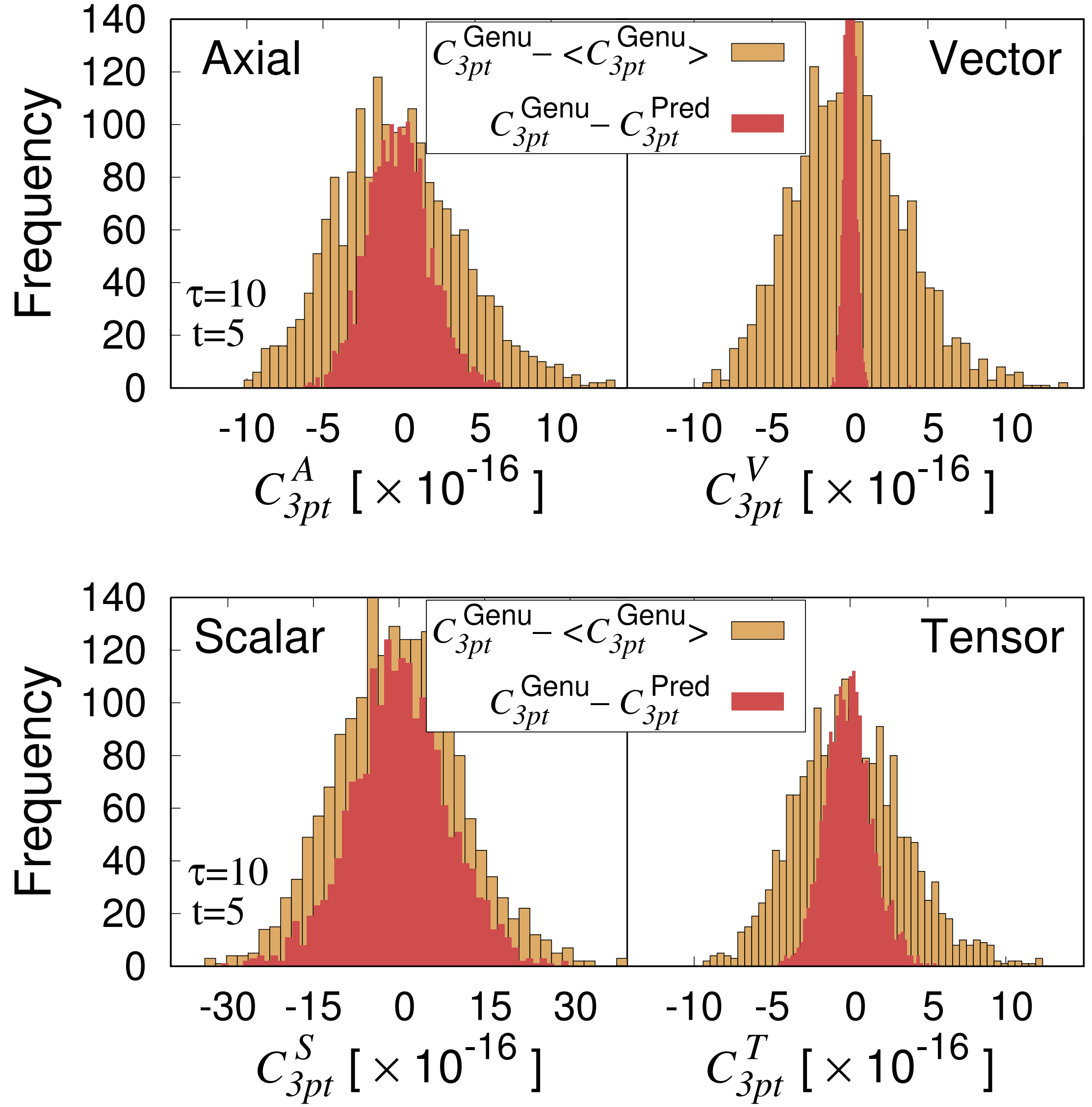
Decision Tree h_{30} for $C^{A}_{3pt}(10, 5)$



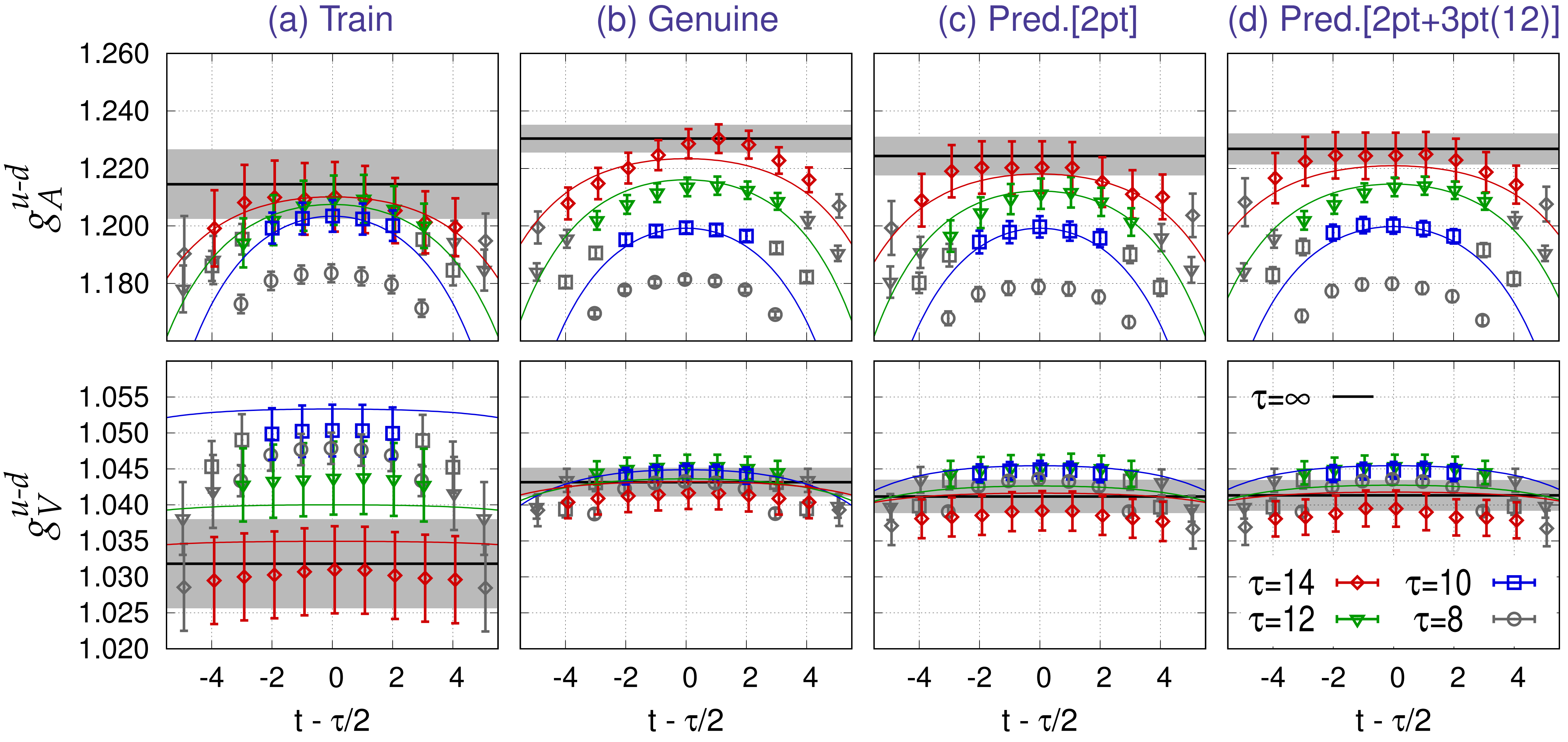
Prediction of C_{3pt} from C_{2pt}

- Training and Test performed for
  - Clover-on-HISQ
  - $a = 0.089\text{fm}$, $M_\pi = 313$ MeV
  - Measurements: 2263 confs × 64 srcs
- \# of Training data: 400 confs
  \# of Test data: 1864 confs
- Predictions of $C^{\Gamma}_{3pt}(10{,}5)/\langle C_{2pt}(10)\rangle$

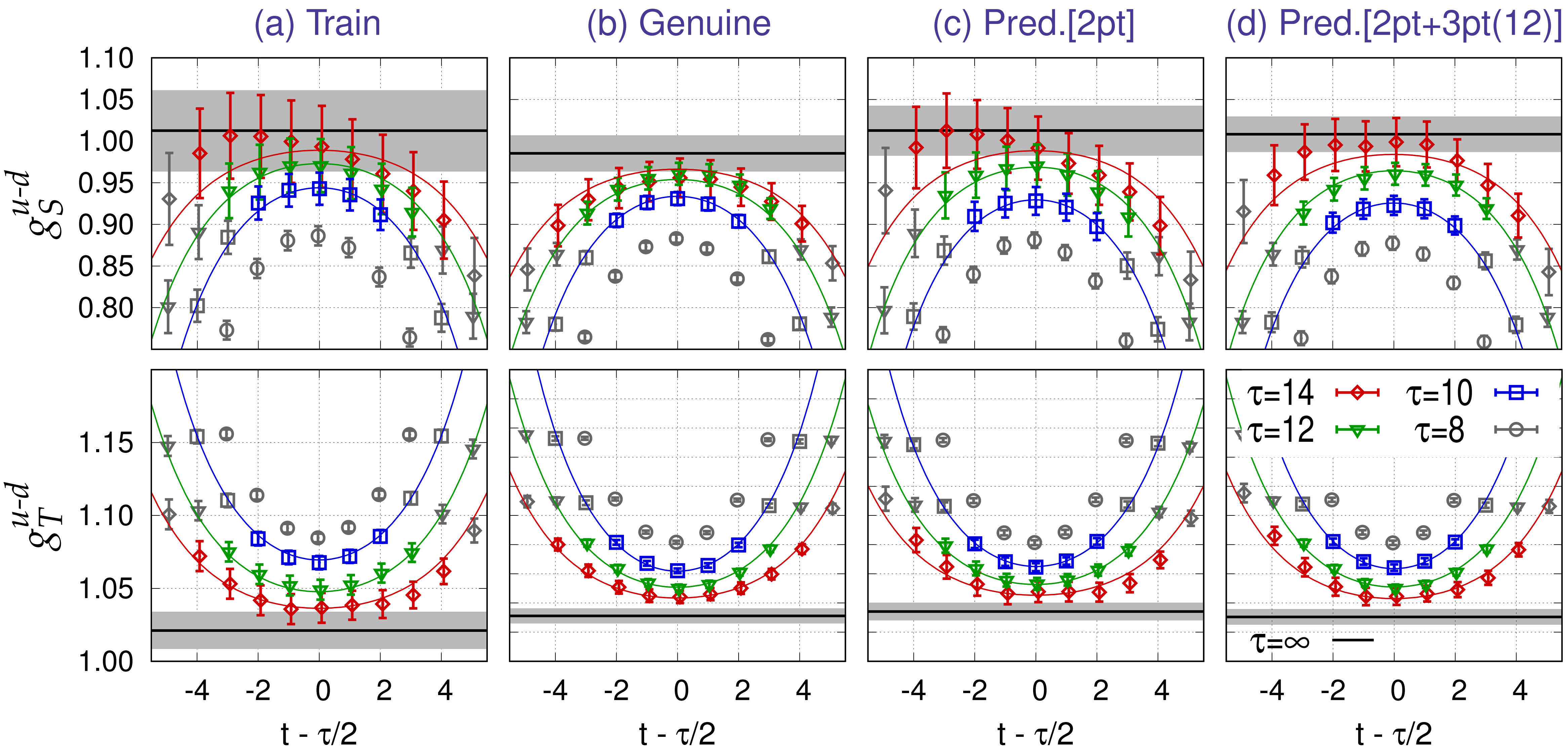
Γ	Genuine	Raw-Prediction	BC-Prediction	Bias
S	0.927(10)	0.921(7)	0.924(20)	+0.003(19)
A	1.1968(38)	1.1971(32)	1.1974(55)	+0.0003(44)
T	1.0594(31)	1.0628(27)	1.0624(40)	−0.0004(30)
V	1.0418(33)	1.0419(32)	1.0422(33)	+0.0003(7)



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



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



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

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

	Genuine	Pred.$[C_{2\mathrm{pt}}]$	Pred.$[C_{2\mathrm{pt}}+C_{3\mathrm{pt}}(12)]$
g_S	0.985(22)	1.013(30)	1.008(21)
g_A	1.2304(48)	1.2243(67)	1.2268(54)
g_T	1.0312(52)	1.0342(61)	1.0304(54)
g_V	1.0432(20)	1.0412(23)	1.0413(21)
	2263 DM (Direct Meas.)	400 DM + 1863 Pred.	400 DM + 1863 Pred.

Quark Chromo EDM (cEDM)

- Simulation in presence of CPV cEDM interaction

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

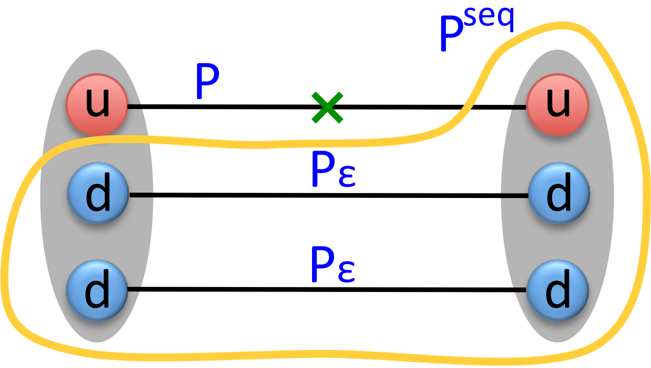
$$S_{cEDM} = -\frac{i}{2}\int d^4x \ \tilde{d}_q g_s \bar{q}(\sigma \cdot G)\gamma_5 q$$

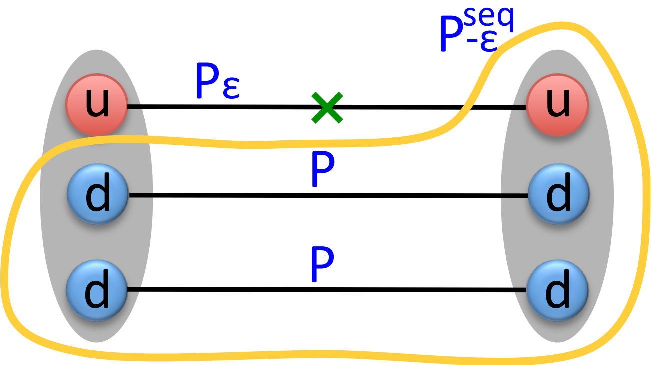
- Schwinger source method

  Include cEDM term in valence quark propagators by modifying Dirac operator

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

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





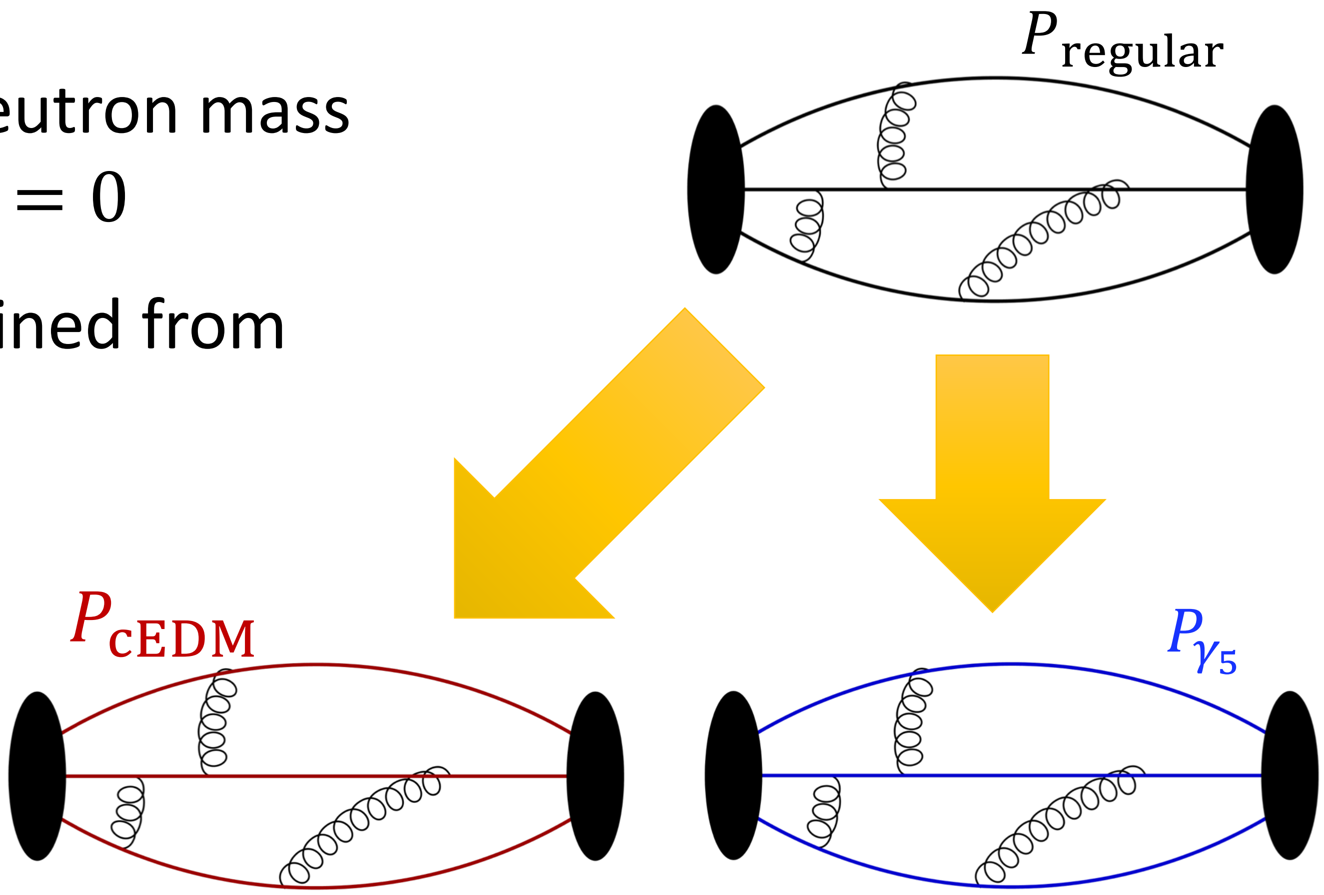
Prediction of $\boldsymbol{C_{2pt}^{\mathrm{CPV}}}$ from $\boldsymbol{C_{2pt}}$

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

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

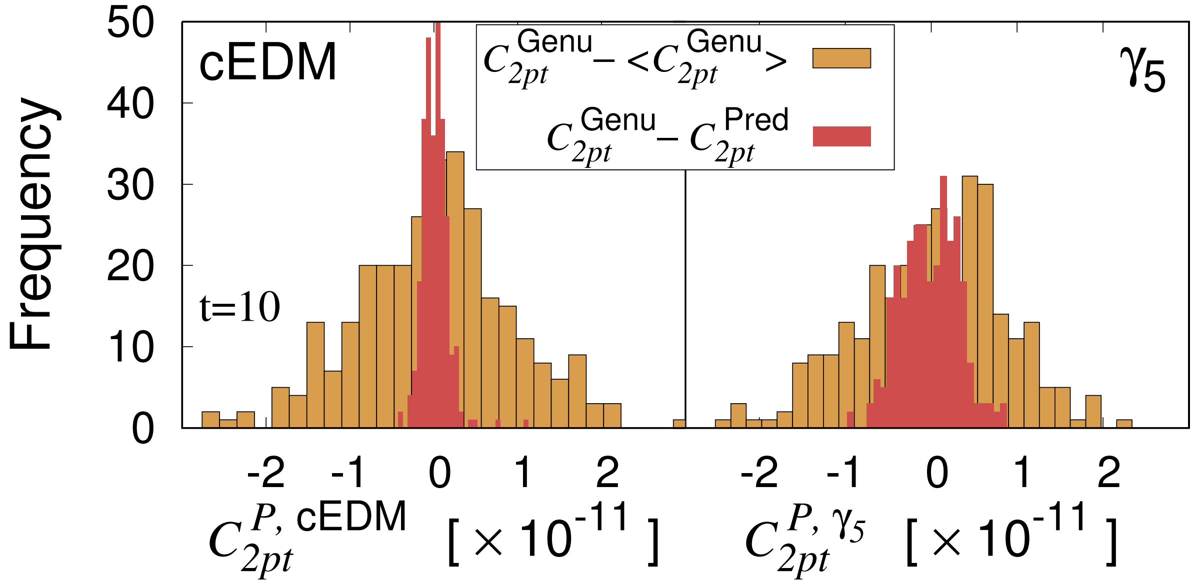
- At leading order, α can be obtained from

$$C_{2pt}^{P} \equiv \mathrm{Tr}(\gamma_5 \langle N \bar{N} \rangle)$$



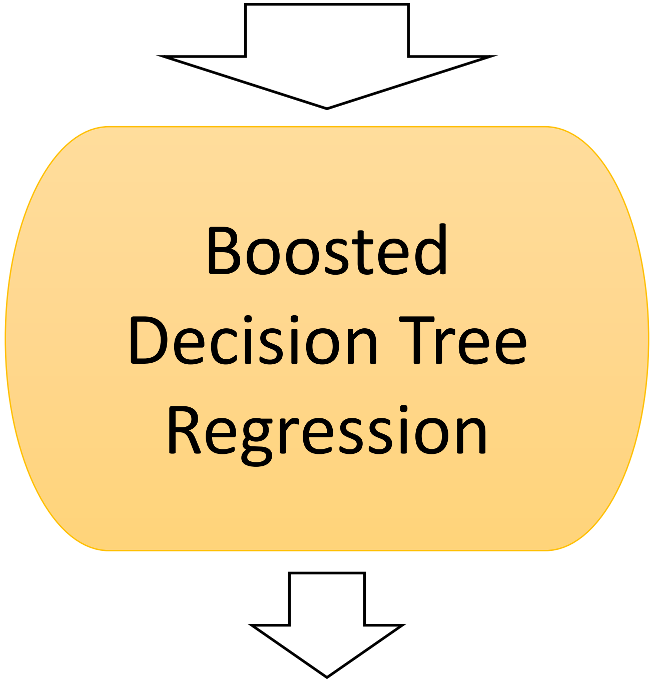
Prediction of C_{2pt}^{CPV} from C_{2pt}

- Training and Test performed for
  - Clover-on-HISQ
  - $a = 0.12$ fm, $M_\pi = 305$ MeV
  - Measurements: 400 confs × 64 srcs
- # of Training data: 100 confs
  # of Test data: 300 confs



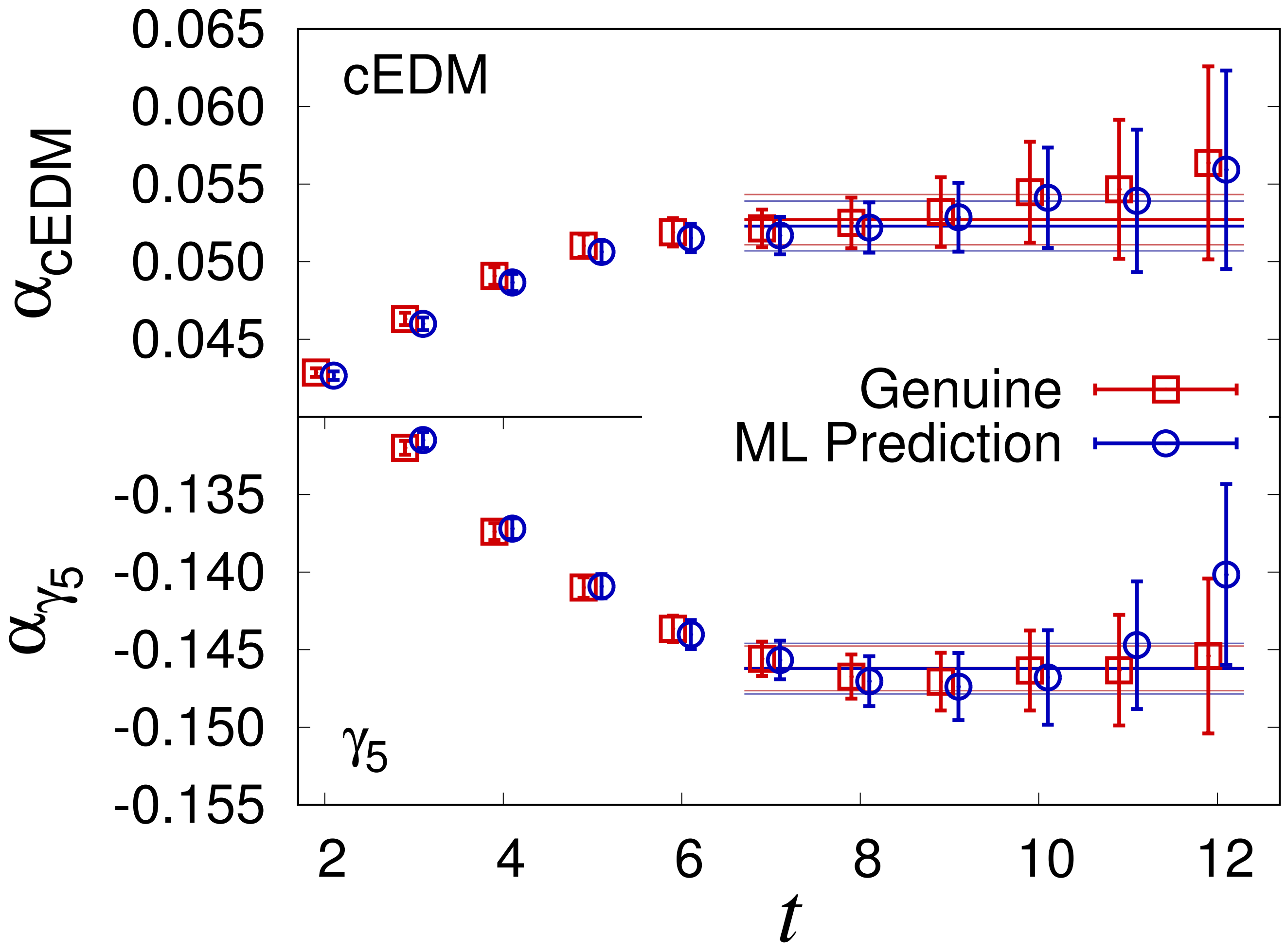
Input:

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



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

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



- $\boldsymbol{\alpha}_{\mathbf{cEDM}}$
  - Genuine: 0.0527(16)
  - Prediction: 0.0523(16)
- $\boldsymbol{\alpha}_{\boldsymbol{\gamma_5}}$
  - Genuine: -0.1462(14)
  - Prediction: -0.1462(16)

➢ Genuine: DM on 400 confs

➢ Prediction: DM on 100 confs + ML prediction on 300 confs

Summary

- Machine learning is used to predict unmeasured observables from measured observables
- Unbiased estimator using cross-validation is presented
- Demonstrated for two lattice QCD calculations:
  1) Prediction of C_{3pt} from C_{2pt}
  2) Prediction of C_{2pt}^{CPV} from C_{2pt}
- The approach can be applied to various lattice calculations and reduce measurement cost

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])
```

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

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