Machine Learning Approximated Nucleon Matrix Elements with Domain Wall Fermions

LATTICE2021@MIT (Virtual)

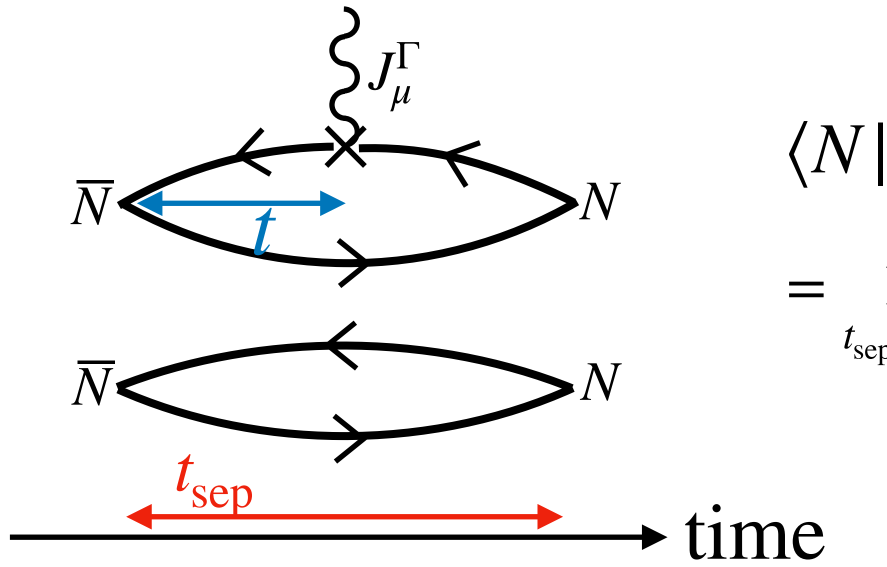
Akio Tomiya (International professional university of technology in Osaka, Assistant Professor)
akio@yukawa.kyoto-u.ac.jp

Joseph Carolan, Connelly Andrew, Taku Izubuchi, Luchang Jin, Chulwoo Jung, Christopher Kelly, Meifeng Lin, Sergey Syritsyn

1. Introduction

(1)

Nucleon charges are important but expensive



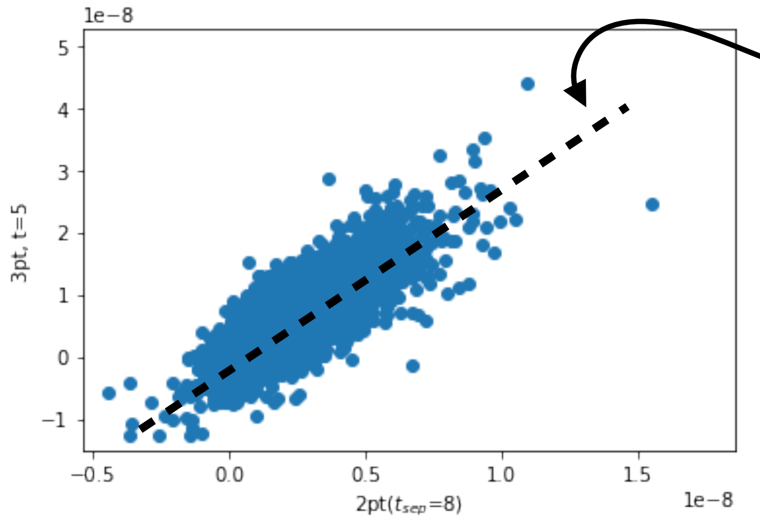
$$\langle N | J^{\Gamma}_{\mu} | N \rangle = \lim_{t_{sep} \gg t \gg 0} \frac{C_{3pt}(\Gamma, t, t_{sep})}{C_{2pt}(t_{sep})}$$

- A lot of contractions, inversions are needed
- 2pt is relatively easy
- B. Yoon et al. [1] used machine learning techniques to predict 3pt functions from 2pt functions with bias corrections
- We apply their method to Domain-wall fermion (DWF) data [2], which has better symmetry.

2. Machine learning?

(2)

- Data determines an approximate map (function) ~ Fitting.



$$F^{\theta}_{\text{app}} : C_{2pt} \mapsto C_{3pt}$$

- θ = a set of parameters
- Tuned to reproduce output (training): $\theta \to \theta^*$
- Approximation birings a bias

- We examine, linear regressior (LR), boosted decision tree (BDT) regressor in scikit-learn library. (Others can be done similar way)
- First we determine parameters, and calculate 3pt functions with fixed parameter
- Above plot suggests, better conservation law gives better results?

3. Bias correction

(3)

Trained machine brings bias, which can be canceled by bias correction term [1, 3],

$$\overline{C}_{3pt} = \frac{1}{N_{\text{Prod}}} \sum_{c=1}^{N_{\text{Prod}}} F^{\theta^*}_{\text{app}}(C^c_{2pt}) + \left[\frac{1}{N_{\text{BC}}} \sum_{c=1}^{N_{\text{BC}}} C^c_{3pt} - \overline{F}^{\theta^*}_{\text{app}}(C_{2pt}) \right]$$

ML approximated
Input 2pt
= Cheap

Bias correction
Input 2pt & 3pt
= Expensive

- If we take expectation value, it becomes exact
- We divide data in following way

Production N_{Prod}	BC N_{BC}	Training N_{Tr}
2pt only	2pt& 3pt	

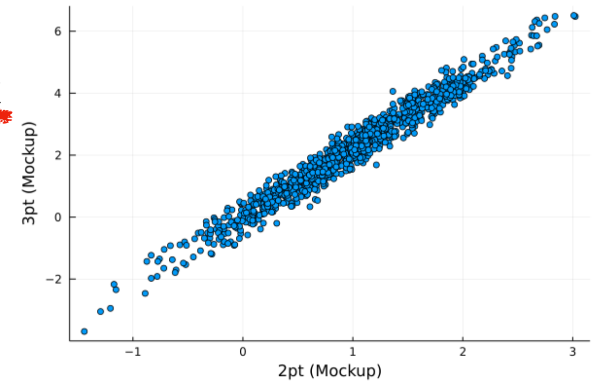
If total error is small under $N_{\text{Prod}} > N_{\text{BC}} + N_{\text{Tr}}$, we get gain

4. Error evaluation: Super-jackknife

(4)

- B. Yoon er al [1] used the bootstrap
- We use super-jackknife. Each term are calculated by Jackknife

$$\delta \mathcal{O}^2_{\text{total}} = \delta \mathcal{O}^2_{\text{Production}} + \delta \mathcal{O}^2_{\text{BC}}$$



5. Mockup data, result (1)

- $X = 1.0 + \eta_1, \ \eta_1 \sim \mathcal{N}(0, 0.5)$
- $y = \alpha X + \eta_2, \ \eta_2 \sim \mathcal{N}(0, 0.1), \alpha = 2.2; \ \ \langle y \rangle = 2.2$
- To examine, we evaluate error by repeating independent sampling

```
Ntr = 200, NBC = 300, Nprod = 5000
```

Model	# of rep	Mean	Std of mean	δProd	δBC	δtotal
Linear Reg.	100	2.200788	0.027386	0.021980	0.018366	0.028643
Linear Reg.	1000	2.199523	0.028720	0.022006	0.018343	0.028648
Linear Reg.	10000	2.199757	0.028185	0.022013	0.018322	0.028640

Consistent! = Correctly evaluated

6. Results for Mockup data (2)

(5)

Model	# of rep	Mean	Std of mean	δProd	δBC	δtotal
BDT	100	2.201785	0.031220	0.021855	0.021431	0.030609
BDT	1000	2.198973	0.029970	0.021913	0.021295	0.030556
BDT	10000	2.200112	0.030557	0.021889	0.021348	0.030576

- Both linear reg. and BDT cases, error is correctly evaluated
- In particular, error from training is not appeared
- Even data is non-linear case, this methodology works (skipped)

7. Lattice setup

- DWF on L = 24^3 x 64 x 16, m = 0.005, mpi ~ 330 MeV, Iwasaki gauge action beta = 2.13 [2].
- 200 configs (5 skipped), 64 measurements on each configurations.
- Input, all time separation of 2pt (t_sep = 0,1,2,…, 18), and determine C3pt (t = 8)
- 80% production data: 10%(Training): 10%(BC) = 10240: 1280: 1280
- Examine 3 point function for the vector channel

8. Results for actual data (Preliminary)

(6)

- Actual results for 3pt (N_{data}= 12800 is used) = 0.140 +/- 0.002
- LR ($N_{\text{BC}} + N_{\text{Tr}}$ = 2560) = 0.140 +/- 0.002, ($\delta\mathcal{O}_{\text{BC}}$:0.0017, $\delta\mathcal{O}_{\text{prod}}$: 0.0096)
- BST ($N_{\text{BC}} + N_{\text{Tr}}$ = 2560) = 0.140 +/- 0.002, ($\delta\mathcal{O}_{\text{BC}}$:0.0018, $\delta\mathcal{O}_{\text{prod}}$: 0.0096)
- # of data for 3pt function is small but error is smaller than the original evaluation

9. Summary

- 3 point functions are correctly reproduced and error are correctly evaluated
- Future: Finite momentum (=form factor), Different channels
- How large gain is?

Reference

1. B. Yoon et al., https://arxiv.org/abs/1807.05971
2. Y.Aoki et al., https://arxiv.org/abs/1011.0892
3. Eigo Shintani et al., https://arxiv.org/abs/1402.0244