

The logo for Jefferson Lab features the text "Jefferson Lab" in white, bold, sans-serif font against a black rectangular background. A red, stylized orbital path with a small sphere at one end curves around the text.

Jefferson Lab

Universal Monte Carlo Event Generator

Supported by Jefferson Lab Laboratory
research and development (LDRD19-13)

Nobuo Sato

CHEP19, Adelaide

Partnership with computer scientists

- Y. Alanazi (ODU)
- M. P. Kuchera (Davidson College)
- Y. Li (co-PI) (ODU)
- T. Liu (JLab)
- R. E. McClellan (JLab)
- W. Melnitchouk (PI) (JLab)
- E. Pritchard (Davidson College)
- R. Ramanujan (Davidson College)
- M. Robertson (Davidson College)
- NS (co-PI) (JLab)
- R. R. Strauss (Davidson College)
- L. Velasco (Dallas)



The big picture

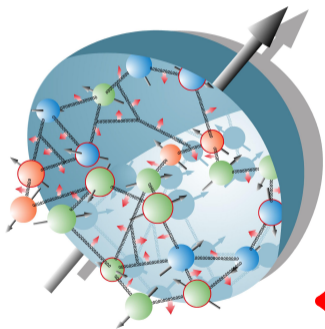
hadrons as **emergent phenomena** of QCD



quarks and gluons

The big picture

hadrons as **emergent phenomena** of QCD



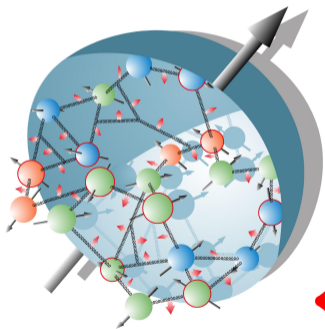
nucleon structure



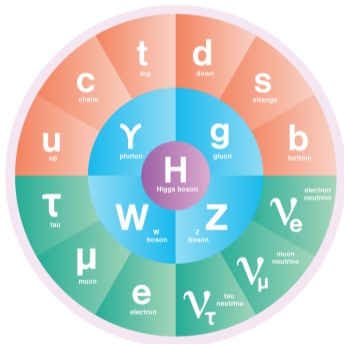
quarks and gluons

The big picture

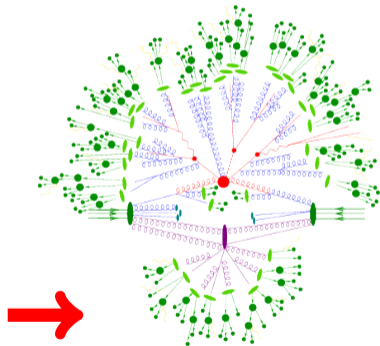
hadrons as **emergent phenomena** of QCD



nucleon structure



quarks and gluons



hadronization

Motivations

- A **new era** of nuclear physics has started with the JLab 12 GeV program

Motivations

- A **new era** of nuclear physics has started with the JLab 12 GeV program
- New tools based on Machine Learning (ML) **to boost the discovery potential** are needed

The goals

The goals

- Build a **theory-free** MCEG

The goals

- Build a **theory-free** MCEG
- Map out particles correlations without biases from **approximated theory**

The goals

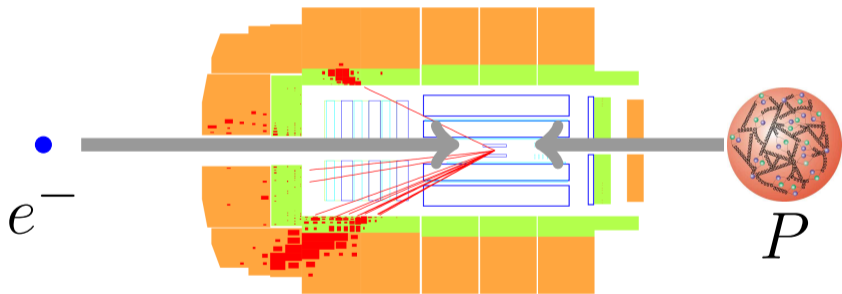
- Build a **theory-free** MCEG
- Map out particles correlations without biases from **approximated theory**
- MCEG as a **data storage utility**

Nature



Nature

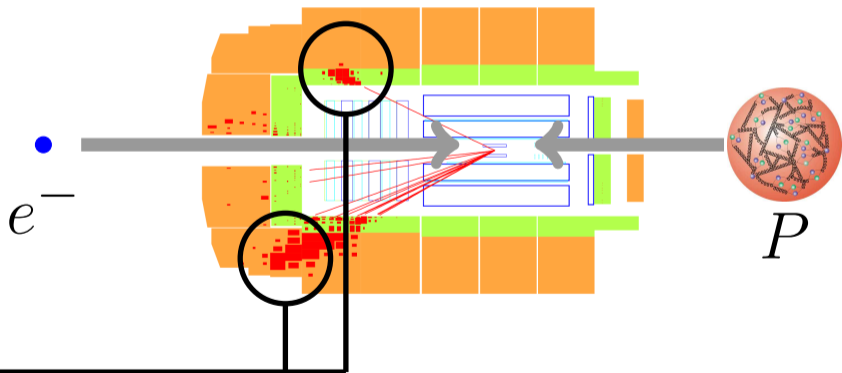
experimental
detector



Nature

experimental
detector

detector level
events



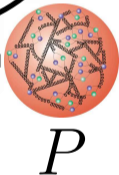
Nature

vertex level
events

experimental
detector

detector level
events

e^{-}



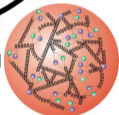
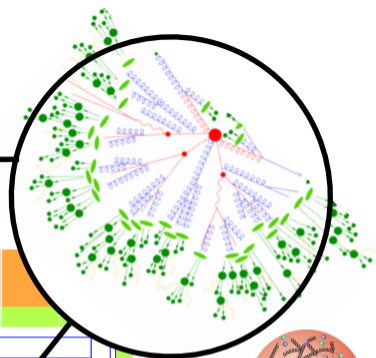
Nature

vertex level
events

experimental
detector

detector level
events

e^{-}



P

Nature

```
graph TD; A[Nature] --> B[vertex level events]; B --> C[experimental detector]; C --> D[detector level events];
```

A vertical flowchart with four rectangular boxes connected by orange arrows pointing downwards. The first box is red and contains the word 'Nature'. The second box is blue and contains 'vertex level events'. The third box is green and contains 'experimental detector'. The fourth box is blue and contains 'detector level events'.

vertex level
events

experimental
detector

detector level
events

Can we use ML to:

Nature

vertex level
events

experimental
detector

detector level
events

Nature

```
graph TD; A[Nature] --> B[vertex level events]; B --> C[experimental detector]; C --> D[detector level events];
```

A vertical flowchart with four rectangular boxes connected by orange arrows pointing downwards. The first box is red and contains the word 'Nature'. The second box is blue and contains 'vertex level events'. The third box is green and contains 'experimental detector'. The fourth box is blue and contains 'detector level events'.

vertex level
events

experimental
detector

detector level
events

Can we use ML to:

- simulate vertex level events?

Nature

```
graph TD; A[Nature] --> B[vertex level events]; B --> C[experimental detector]; C --> D[detector level events];
```

The diagram illustrates a four-stage process. It begins with a red box labeled 'Nature'. An orange arrow points down to a blue box labeled 'vertex level events'. Another orange arrow points down to a green box labeled 'experimental detector'. A final orange arrow points down to a blue box labeled 'detector level events'.

vertex level
events

experimental
detector

detector level
events

Can we use ML to:

- simulate vertex level events?
- simulate detector level events?

```
graph TD; A[Nature] --> B[vertex level events]; B --> C[experimental detector]; C --> D[detector level events];
```

Nature

vertex level
events

experimental
detector

detector level
events

Can we use ML to:

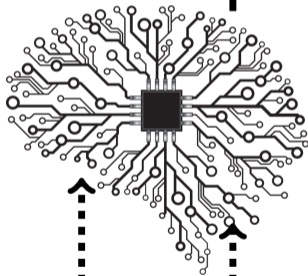
- simulate vertex level events?
- simulate detector level events?
- simulate **nature**?

Nature

vertex level
events

experimental
detector

detector level
events



UMCEG

vertex level
events

detector
simulator

detector level
events

Nature

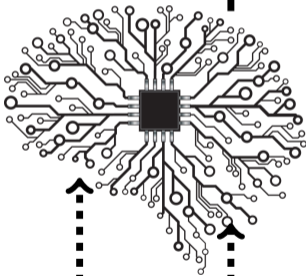
data compression

UMCEG

vertex level
events

experimental
detector

detector level
events



vertex level
events

detector
simulator

detector level
events

Our strategy

- Event level ML training → **GAN**

Our strategy

- Event level ML training \rightarrow **GAN**
- Use a dual GAN as the event generator

$$\underbrace{\rho(\text{particles}|\text{multiplicity})}_{\text{vectors generator}} \times \underbrace{\rho(\text{multiplicity})}_{\text{multiplicity generator}}$$

Challenges

- Find optimal data representation
→ what is the **image of an event**?

Challenges

- Find optimal data representation
→ what is the **image of an event**?
- How to make the GAN to learn the **features of the event**? → CNN

Challenges

- Find optimal data representation
→ what is the **image of an event**?
- How to make the GAN to learn the **features of the event**? → CNN
- How to escalate from low to higher multiplicities?

Our current work in progress

- Use Pythia as a training and validation tool

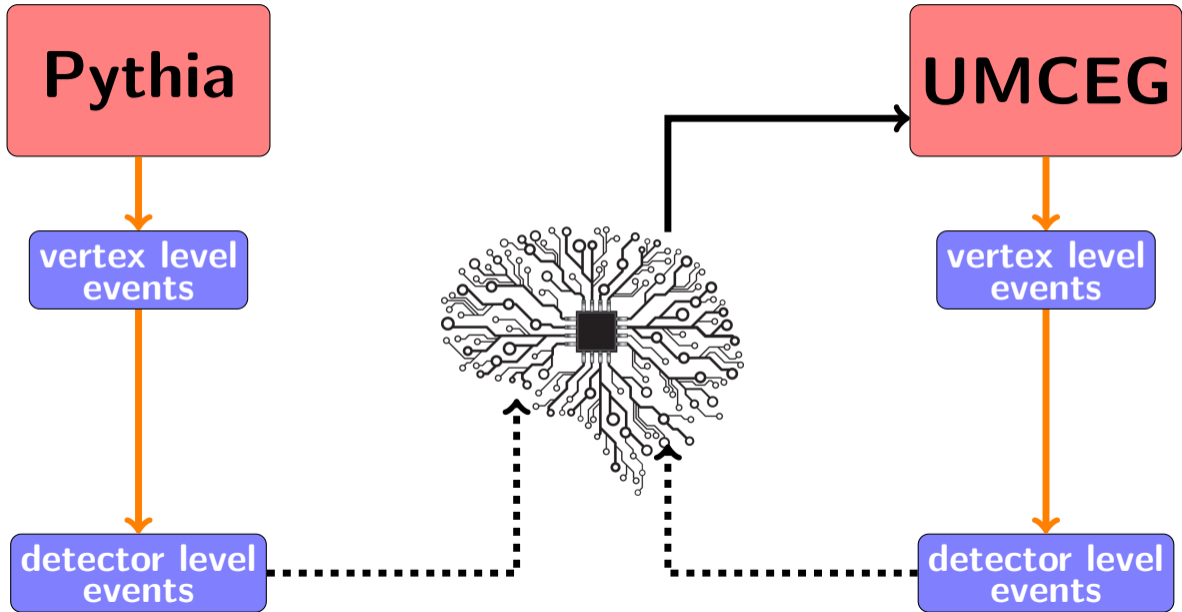
Our current work in progress

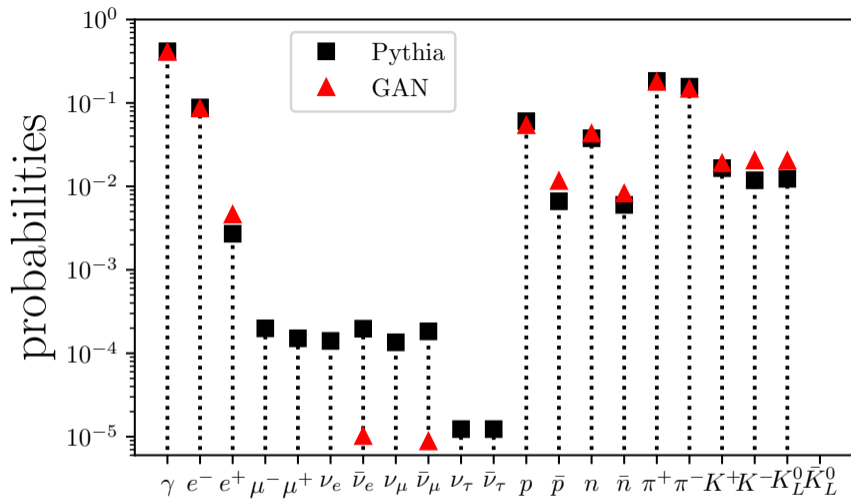
- Use Pythia as a training and validation tool
- Ignore detector effects

Our current work in progress

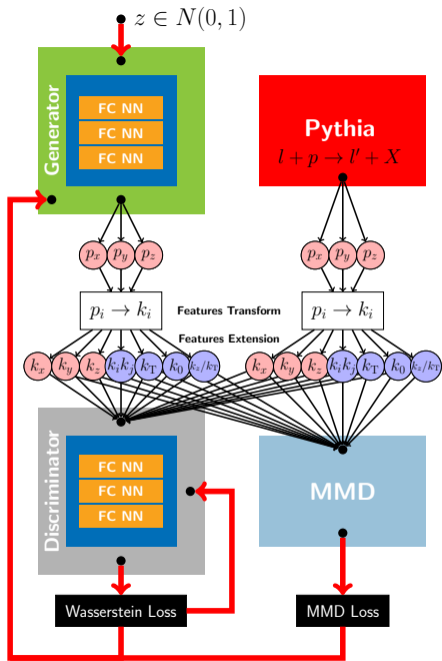
- Use Pythia as a training and validation tool
- Ignore detector effects
- Start with inclusive particle generator

$$\rho(\text{particles}|\text{multiplicity}) \rightarrow \rho(\text{particles} + X)$$

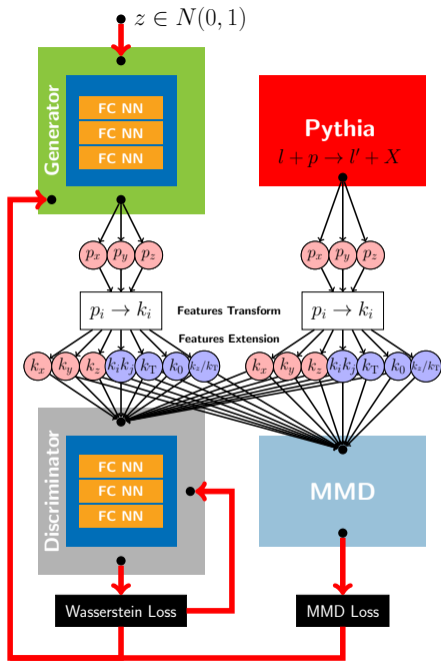




Multiplicity generator

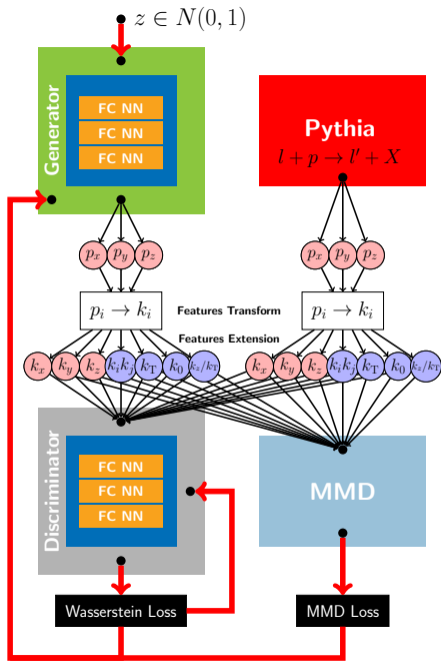


Vectors generator



■ Event image = $l'_{x,y,z}$

Vectors generator

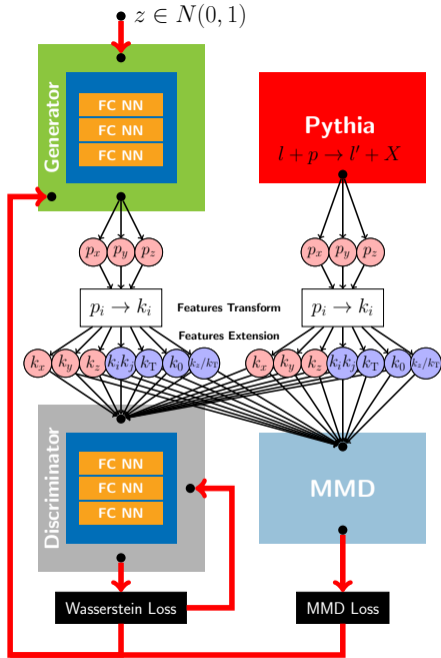


■ Event image = $l'_{x,y,z}$

■ Feature extension:

$$l'_i \cdot l'_j, \quad l'_0, \quad l'_z/l'_T$$

Vectors generator



■ Event image = $l'_{x,y,z}$

■ Feature extension:

$$l'_i \cdot l'_j, \quad l'_0, \quad l'_z/l'_T$$

■ WGAN+MMD Butter, Plehn, Winterhalder ('19)

Vectors generator

Validation

Validation

- Relevant observables for inclusive DIS

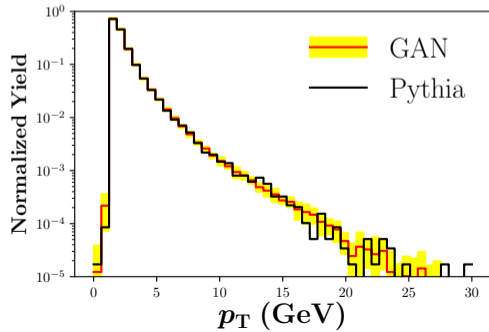
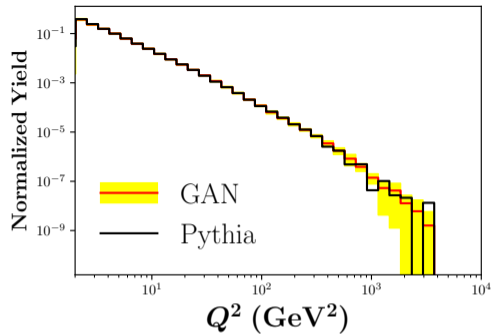
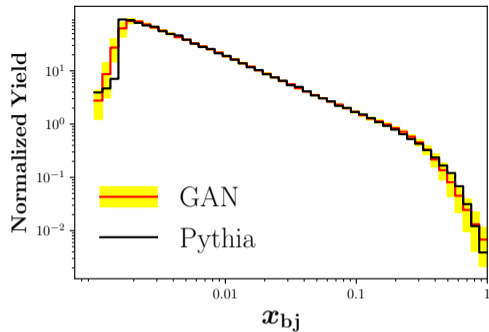
$$Q^2 = -(l - l')^2 \quad x_{\text{bj}} = \frac{Q^2}{2P \cdot (l - l')}$$

Validation

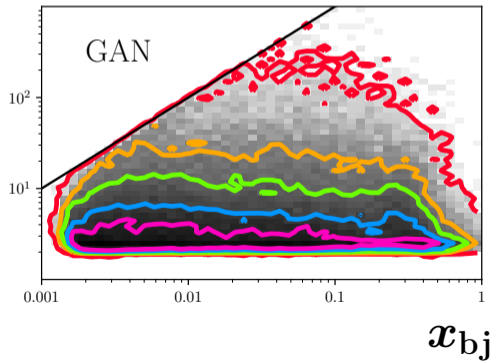
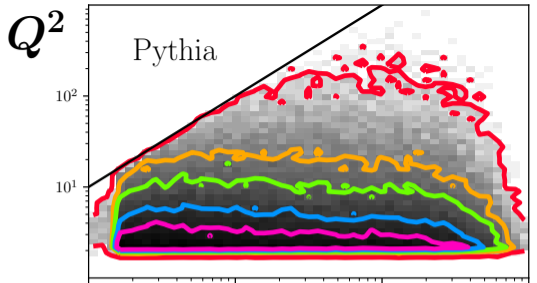
- Relevant observables for inclusive DIS

$$Q^2 = -(l - l')^2 \quad x_{\text{bj}} = \frac{Q^2}{2P \cdot (l - l')}$$

- x_{bj}, Q^2 **not included** as features



Error bands generated with bootstrapped samples



- Isocontours are in agreement
- x_{bj}, Q^2 correlation is learned without adding $x_{bj} \cdot Q^2$ feature

Summary and outlook

- It is possible to train a GAN at the **event level** to build a MCEG

Summary and outlook

- It is possible to train a GAN at the **event level** to build a MCEG
- The current design provides **a blueprint** for a generator with higher multiplicity

Summary and outlook

- More work is needed, but the **results are encouraging**

Summary and outlook

- More work is needed, but the **results are encouraging**
- A fully trained UMCEG will be a **complementary tool** to theory-based MCEGs such as Pythia