Deep-Learning Unfolding for Extraction of Drell-Yan Angular Parameters in pp Collisions with 120 GeV Beam Energy

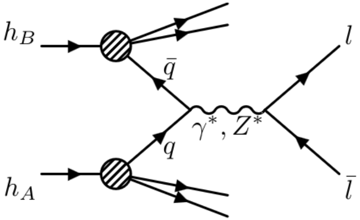
Dinupa Nawarathne

New Mexico State University
Representing the FermiLab SeaQuest/E906 Collaboration

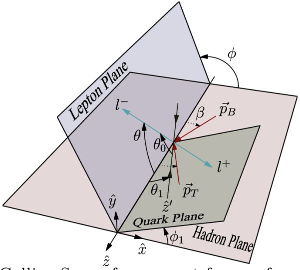
APS-DNP-2024, October 10, 2024

Drell-Yan Process

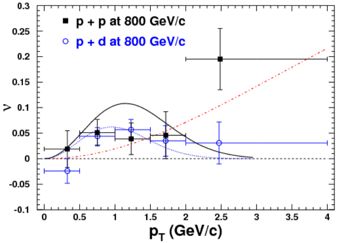
$$\frac{d\sigma}{d\Omega} = 1 + \lambda \cos^2\theta + \mu \sin 2\theta \cos\phi + \frac{1}{2}\nu \sin^2\theta \cos 2\phi$$



Drell-Yan process[1]



Collins-Soper frame: rest frame of the dileptons[2]



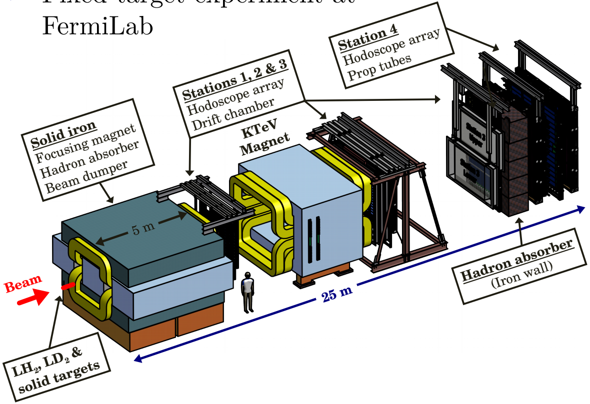
Extraction of ν parameter in FermiLab-E866/NuSea experiment[3]. Curves are theoretical calculations.[4]

- λ, μ and ν - angular coefficients of the lepton angular distribution.
- Non-zero ν parameter provide information about the transverse motion of the quarks inside the proton.

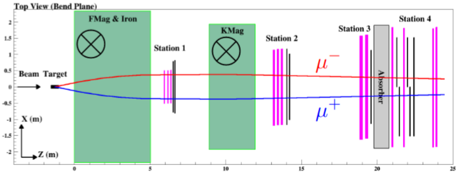
[1] F. Bechtel, PhD thesis, Hamburg U., 2009.
[2] J.-C. Peng *et al.*, *Phys. Lett. B* **789**, 356–359, arXiv: 1808.04398 (hep-ph) (2019).
[3] L. Y. Zhu *et al.*, *Phys. Rev. Lett.* **102**, 182001, arXiv: 0811.4589 (nucl-ex) (2009).
[4] B. Zhang *et al.*, *Phys. Rev. D* **77**, 054011, (https://link.aps.org/doi/10.1103/PhysRevD.77.054011) (5 2008).

FermiLab SeaQuest/E906 Experiment

- Fixed target experiment at FermiLab



SeaQuest spectrometer[5]



$\mu^{+/-}$ tracks[6]

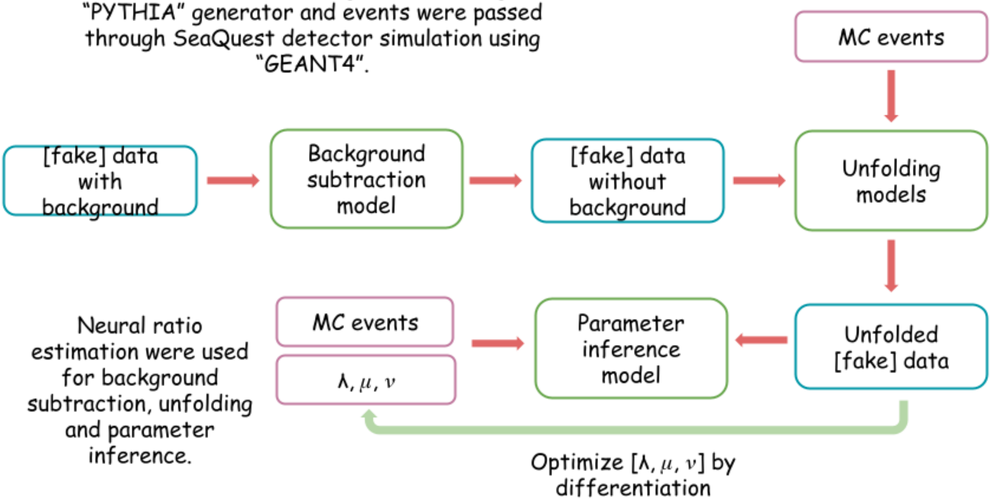
- Use 120 GeV beam energy from main injector.
- Main goal is to measure the antiquark structure of the nucleon.
- SeaQuest data is capable of extracting the Drell-Yan angular coefficients.

[5] C. A. Aidala *et al.*, *Nucl. Instrum. Meth. A* **930**, 49–63, arXiv: 1706.09990 (physics.ins-det) (2019).

[6] K. Nagai, PhD thesis, Tokyo Inst. Tech, 2017.

Analysis Pipeline

Monte-Carlo data was generated using "PYTHIA" generator and events were passed through SeaQuest detector simulation using "GEANT4".



Neural ratio estimation were used for background subtraction, unfolding and parameter inference.

Neural Likelihood Ratio Estimation

- Consider two joint probability distributions of the Drell-Yan process $p(\phi, \cos\theta|\lambda_1, \mu_1, \nu_1)$ and $p(\phi, \cos\theta|\lambda_0, \mu_0, \nu_0)$. Then the likelihood ratio is given by,

$$\mathcal{L} = \frac{p(\phi, \cos\theta|\lambda_0, \mu_0, \nu_0)}{p(\phi, \cos\theta|\lambda_1, \mu_1, \nu_1)}$$

- Let $f(\phi, \cos\theta)$ be a classifier trained to classify between two equal-sized samples $\phi, \cos\theta \sim p(\phi, \cos\theta|\lambda_0, \mu_0, \nu_0)$, labeled $y = 0$ and $\phi, \cos\theta \sim p(\phi, \cos\theta|\lambda_1, \mu_1, \nu_1)$, labeled $y = 1$. Then likelihood ratio estimator can be defined as;

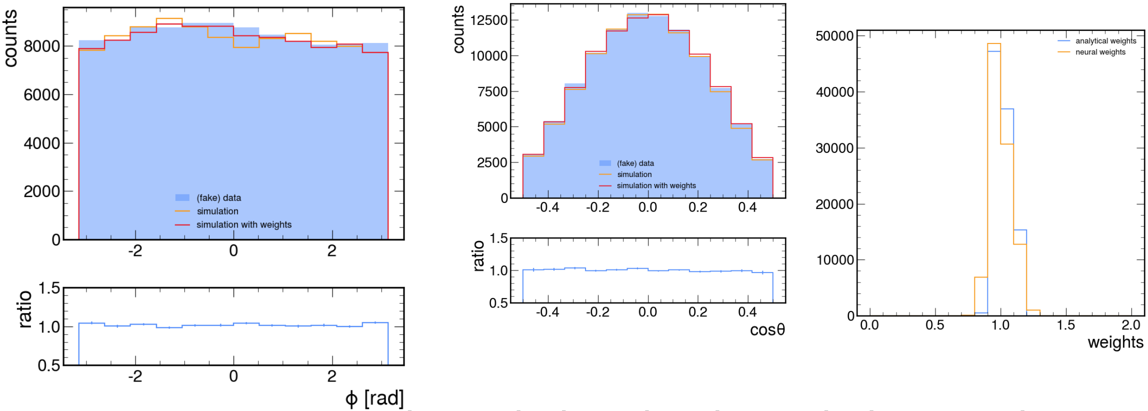
$$\hat{r}(\phi, \cos\theta) = \frac{1 - f(\phi, \cos\theta)}{f(\phi, \cos\theta)} \tag{1}$$

- We can use equation 1 as a reweighting method;

$$p(\phi, \cos\theta|\lambda_0, \mu_0, \nu_0) \approx \hat{r}(\phi, \cos\theta) p(\phi, \cos\theta|\lambda_1, \mu_1, \nu_1)$$

  i.e. even if the analytical formula for the likelihood ratio is intractable, we can approximate the likelihood ratio between two joint probability distributions.[7]

[7] J. Brehmer *et al.*, *Phys. Rev. D* **98**, 052004, arXiv: 1805.00020 (hep-ph) (2018).



In this example we choose values $[\lambda_1, \mu_1, \nu_1] = [1., 0., 0]$ and $[\lambda_0, \mu_0, \nu_0] = [0.9, -0.2, 0.1]$. Reconstructed ϕ and $\cos\theta$ are used as inputs to the neural network. The weights calculated using the neural ratio estimator show good agreement with the analytically derived weights. We will employ this method for reweighting in background subtraction, unfolding, and parameter inference.[8]

[8] J. Brehmer *et al.*, *Comput. Softw. Big Sci.* **4**, 3, arXiv: 1907.10621 (hep-ph) (2020).

Background Subtraction

- Fist we need to subtract the background[9][10] from the measured (fake) data. Consider the joint probability density of measured (fake) data,

$$p_{\text{meas}}(x|\lambda,\mu,\nu) = p_{\text{(fake) data}}(x|\lambda_0,\mu_0,\nu_0) + p_{\text{background}}(x|\lambda_1,\mu_1,\nu_1)$$

- Then the joint probability density of the (fake) data are given by,

$$p_{\text{(fake) data}}(x|\lambda_0,\mu_0,\nu_0) = p_{\text{meas}}(x|\lambda,\mu,\nu) - p_{\text{background}}(x|\lambda_1,\mu_1,\nu_1)$$

- This can be achieved by minimizing the loss function,[11,12]

$$L[f(x)] = -\sum_{\text{meas}} \log(f(x)) + \sum_{\text{background}} w \log(1-f(x)) - \sum_{\text{meas}} \log(1-f(x))$$

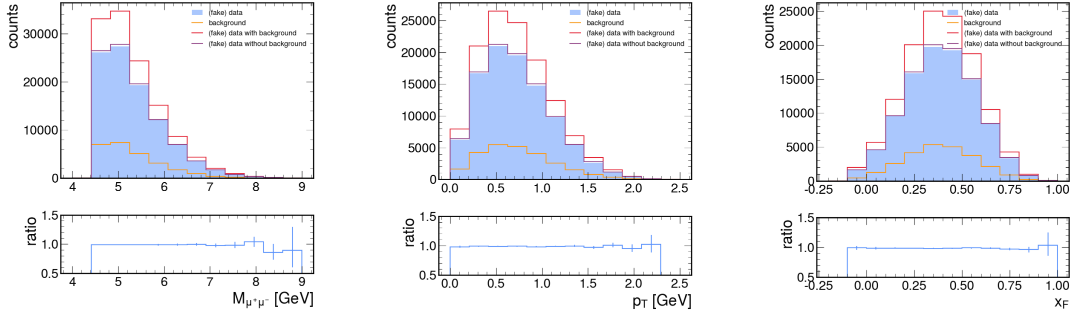
  i.e. in order to achieve the desired loss, we add the measured (fake) data in twice, once with $y=0$ and once with $y=1$ and then we add in the background with a negative weight and $y=0$.

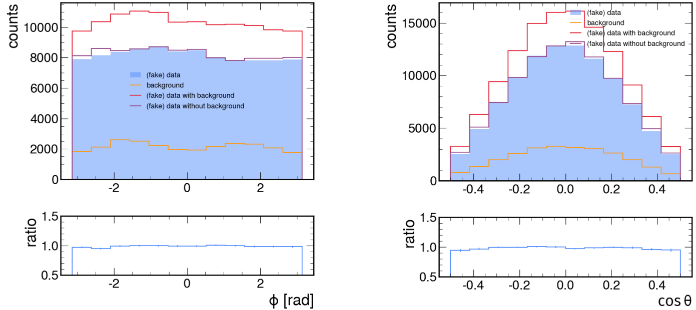
[9] In SeaQuest experiment is dominated by combinatoric background and estimated by event mixing method

[10] S. F. Pate *et al.*, *JINST* **18**, P10032, arXiv: 2302.04152 (hep-ex) (2023).

[11] A. Andreassen, B. Nachman, *Phys. Rev. D* **101**, 091901, arXiv: 1907.08209 (hep-ph) (2020).

[12] A. Andreassen *et al.*, presented at the 9th International Conference on Learning Representations, arXiv: 2105.04448 (stat.ML).





$$\omega = \frac{1 - f(x)}{f(x)}$$

$f(x)$ - binary classifier,
$x = [M_{\mu^+\mu^-}, p_T, x_F, \phi, \cos\theta]$, x - reco. events with background.

Unfolding

- We use the OmniFold[13] algorithm for unfolding (smearing correction).
- This is an iterative Bayesian unfolding algorithm that applies "pushing" and "pulling" weights between detector-level (reconstructed information) and particle-level (true information).
- In the first iteration, simulated reconstructed events are reweighted to match the (fake) data at the reconstruction level by classifying the them. The pulling weights, ω_{pull}, are then calculated as,

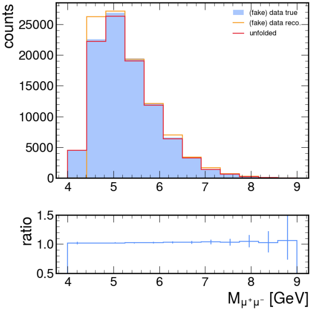
$$\omega_{pull} = \frac{1 - f(x_{sim}^{reco.})}{f(x_{sim}^{reco.})}$$

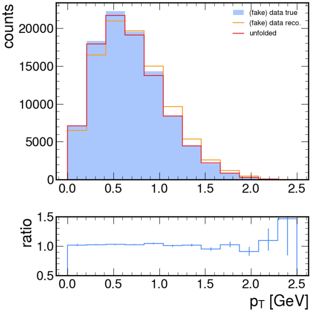
- Next, the simulated true events are reweighted to match the weighted (ω_{pull}) simulated true events by classifying them. The pushing weights, ω_{push}, are then calculated as,

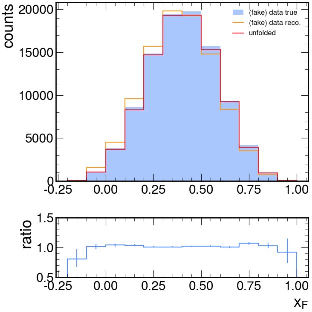
$$\omega_{push} = \frac{1 - f(x_{sim}^{true})}{f(x_{sim}^{true})}$$

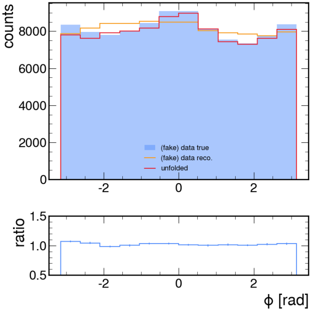
- These ω_{push} and ω_{pull} weights are updated iteratively.

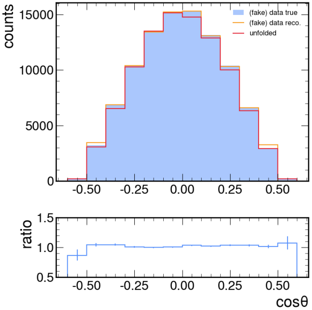
[13] A. Andreassen *et al.*, presented at the 9th International Conference on Learning Representations, arXiv: 2105.04448 (stat.ML).











$$\omega = \frac{1 - f(x_{sim}^{true})}{f(x_{sim}^{true})}$$

$f(x)$ - binary classifier,
$x = [M_{\mu^+\mu^-}, p_T, x_F, \phi, \cos\theta]$,
x_{sim}^{true} - simulated true events.

Parameter Inference

- In this step we want to train a parameterized classifier that can calculate weights for any λ, μ and ν. This is achieved by minimizing the loss,[14,15,16]

$$L[\hat{r}(x|\Theta_0,\Theta_1)] = \frac{1}{N}\sum y\left[r(x|\Theta_0,\Theta_1) - \hat{r}(x|\Theta_0,\Theta_1)\right]^2 + (1-y)\left[\frac{1}{r(x|\Theta_0,\Theta_1)} - \frac{1}{\hat{r}(x|\Theta_0,\Theta_1)}\right]^2$$

where $x = [\phi, \cos\theta]$, $\Theta = [\lambda, \mu, \nu]$, $[\lambda_1, \mu_1, \nu_1] = [1., 0., 0.]$,

$$r(x|\Theta_0,\Theta_1) = \frac{1 + \lambda_0\cos^2\theta + \mu_0\sin 2\theta\cos\phi + \frac{1}{2}\nu_0\sin^2\theta\cos 2\phi}{1+\cos^2\theta}$$

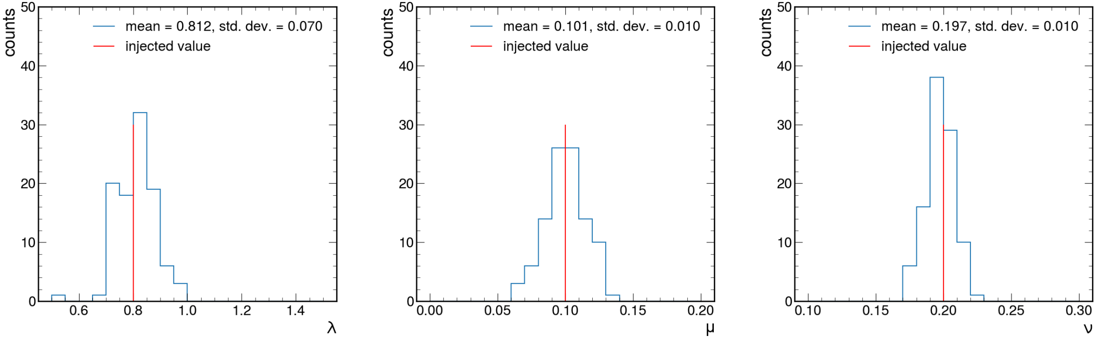
$$\lambda_0 \sim \mathcal{U}(-1., 1.)$$
$$\mu_0 \sim \mathcal{U}(-0.5, 0.5)$$
$$\nu_0 \sim \mathcal{U}(-0.5, 0.5)$$

$$\hat{r}(x|\Theta_0,\Theta_1) = \frac{1 - f(x,\Theta_0,\Theta_1)}{f(x,\Theta_0,\Theta_1)}$$

[14] J. Brehmer *et al.*, *Phys. Rev. D* **98**, 052004, arXiv: 1805.00020 (hep-ph) (2018).
[15] J. Brehmer *et al.*, *Comput. Softw. Big Sci.* **4**, 3, arXiv: 1907.10621 (hep-ph) (2020).
[16] A. Andreassen, B. Nachman, *Phys. Rev. D* **101**, 091901, arXiv: 1907.08209 (hep-ph) (2020).

Parameter Inference

- We can train a classifier of the form $f(x, \Theta)$ by classifying the samples x, Θ_1 with label $y = 1$ and x, Θ_0 with label $y = 0$.
- Because $f(x, \Theta)$ is differentiable, it can be used to directly learn and update these parameters during the inference process, allowing us to infer the best parameter.
- Uncertainties: We found that the variance in the MC events during the training of the parametric model results in large uncertainty. This can be estimated by retraining the parametric model with resampled MC events and using the spread as a source of uncertainty.



Summary

- The Drell-Yan process is an important experimental approach for exploring the partonic structure of nucleons.
- A non-zero ν parameter in the Drell-Yan process provides insights into the transverse motion of quarks within the proton.
- Deep neural networks based classifiers are ideal candidates for likelihood estimators because of their ability to approximate complex, non-linear functions.
- Reweighting joint probability distributions is an excellent method for unbinned background subtraction, unfolding, and parameter inference.
- We are investigating all possible uncertainties in this analysis pipeline.
- Acknowledgement: This work was supported in part by US DOE grant DE-FG02-94ER40847.

Thank You