

# Diffusion models for hadron physics



Università  
degli Studi di  
Messina

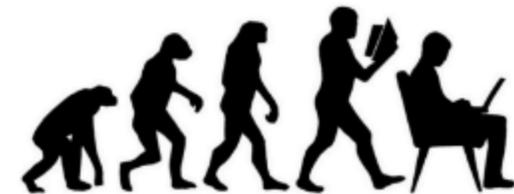


GIORGIO FOTI (UNIME & INFN CATANIA) IN  
COLLABORATION WITH JITAO XU, LUKASZ BIBRZYCKI,  
TOMMASO VITTORINI, TAREQ ALGHAMDI, TONINO FULCI

"AI FOR HADRON SPECTROSCOPY AT JLAB"

JUN 4 - 5, 2025

JEFFERSON LAB



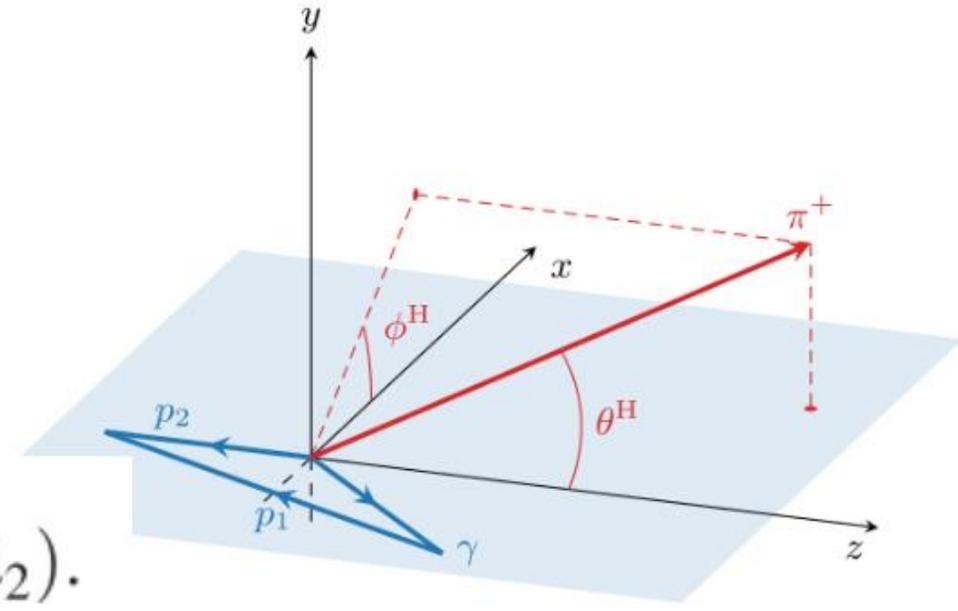
**A(i)DAPT**

**AI for Data Analysis and PreservaTion**

# Process & Kinematics

Let's define the  $\pi\pi$  helicity rest frame for

$$\gamma(q, \lambda_\gamma) + p(p_1, \lambda_1) \rightarrow \pi^+(k_1) + \pi^-(k_2) + p(p_2, \lambda_2).$$



$$E_{\mathbf{p}_1^H} = \frac{s - m_N^2 + t}{2\sqrt{s_{12}}},$$

$$E_{\mathbf{q}^H} = \frac{s_{12} - t}{2\sqrt{s_{12}}},$$

$$\cos \theta_1 = \frac{2m_N^2 - 2E_{\mathbf{p}_1^H}E_{\mathbf{p}_2^H} - t}{2|\mathbf{p}_1^H||\mathbf{p}_2^H|},$$

$$\mathbf{p}_1^H = |\mathbf{p}_1^H|(\sin \theta_1, 0, \cos \theta_1),$$

$$\mathbf{q}^H = |\mathbf{q}^H|(-\sin \theta_q, 0, \cos \theta_q),$$

$$\mathbf{p}_2^H = |\mathbf{p}_2^H|(0, 0, -1),$$

$$E_{\mathbf{p}_2^H} = \frac{s - s_{12} - m_N^2}{2\sqrt{s_{12}}},$$

$$E_{\mathbf{k}_1^H} = \frac{1}{2}\sqrt{s_{12}},$$

$$\cos \theta_q = \frac{s + t - s_{12} - m_N^2 - 2E_{\mathbf{p}_1^H}E_{\mathbf{q}^H}}{2|\mathbf{p}_1^H||\mathbf{q}^H|}.$$

$$\mathbf{k}_1^H = |\mathbf{k}_1^H|(\sin \theta^H \cos \phi^H, \sin \theta^H \sin \phi^H, \cos \theta^H),$$

$$\mathbf{k}_2^H = -|\mathbf{k}_2^H|(\sin \theta^H \cos \phi^H, \sin \theta^H \sin \phi^H, \cos \theta^H),$$

$$E_{\mathbf{k}_2^H} = \frac{1}{2}\sqrt{s_{12}}.$$

# Diffusion Models for Unfolding Detector Effects

- Detector introduce resolution effects, acceptance gaps and inefficiencies that distort the true particle distributions from an experiment. The observed (reconstructed) data distribution  $\text{det}(y)$  is a “smeared” version of the true physics distribution  $\text{true}(x)$ :

$$\text{det}(y) = \int_x P(y|x) \text{true}(x)$$

where  $P(y|x)$  is the detector response.

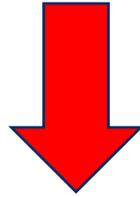
- We seek the inverse of this process – i.e. to infer the posterior probability  $P(x|y)$  or  $\text{true}(x)$  given the measured  $y$  (“**unfolding**”). By Bayes' theorem,

$$P(x|y) = \frac{P(y|x) \text{true}(x)}{\text{det}(y)}$$

- Sample  $x \sim P(x|y)$  to reconstruct the true distribution. However,  $P(x|y)$  depends on the unknown prior  $\text{true}(x)$ . Any prior mismatch can bias the unfolding result. Unfolding is thus an ill-posed inverse problem that we can tackle by introducing additional information or inductive bias.

# Limitations of Traditional Unfolding

- Standard techniques typically require binning data and assume a linear response matrix.
- This limits resolution and prevents simultaneous unfolding of many correlated observables.
- Many algorithms assume a specific underlying distribution, so the unfolded result can be significantly biased.
- Traditional methods also struggle to propagate complex uncertainties.



Need for AI: these challenges motivate AI-based unfolding, which can handle high-dimensional input (e.g. full event kinematics or images) and learn complex detector effects without manual binning.

---

# AI-Based Unfolding Approaches

- Generative models: adversarial and likelihood-based models have been applied: e.g. GANs, (conditional) normalizing flows, can learn to map detector to ground truth distributions (see Tareq's talk).
- Diffusion models: recent work uses diffusion models to represent the unfolding map as a learned generative process. These can, in principle, model complex posteriors and provide uncertainty quantification.

Each AI method has trade-offs; diffusion models are attractive for their strong theoretical foundations and flexibility

Alert: here and in the following the roles of  $x$  and  $y$  are interchanged, because at this stage of the project we have in mind a proof of principle and not the unfolding of the real data

# Denoising Diffusion Probabilistic Models (DMs)

- A DM is a generative model consisting of a *forward diffusion* (noise addition) process and a *learned reverse* (denoising) process. In the forward pass, one gradually corrupts a clean data sample  $\mathbf{x}_0$  by injecting Gaussian noise over  $T$  timesteps, producing  $\mathbf{x}_T \approx N(0, I)$ . In the reverse pass, a neural network learns to iteratively denoise  $\mathbf{x}_T$  back to  $\mathbf{x}_0$ .

- **Reversible generative process:** Formally, the forward (noising) Markov chain is defined as

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = N(\mathbf{x}_t; \sqrt{\alpha_t} \mathbf{x}_{t-1}, (1 - \alpha_t) I)$$

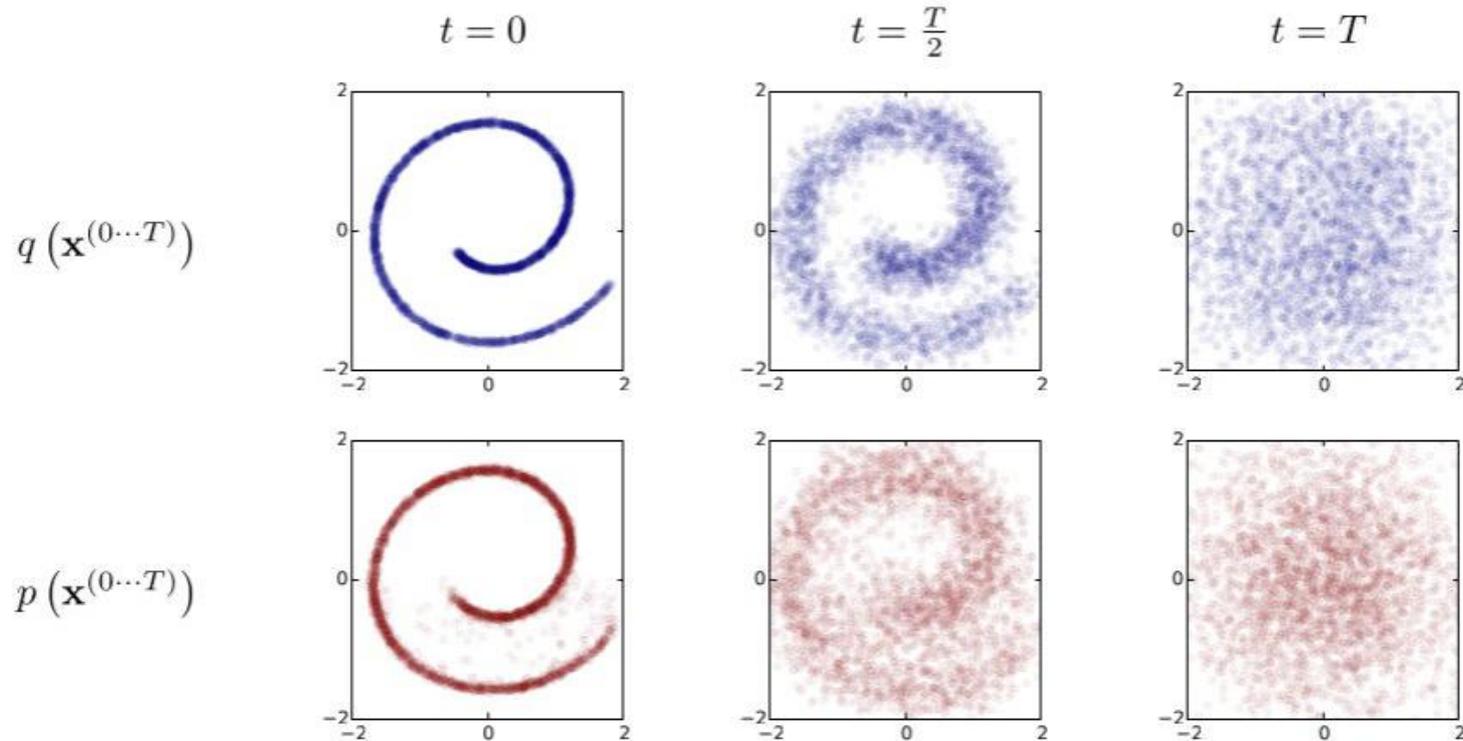
where  $\alpha_t$  is a variance schedule. The learned reverse (denoising) chain is parameterized by  $\theta$ :

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = N(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2 I)$$

- By training the model to predict the added noise, one learns the mean of the Gaussian transition,  $\mu_\theta(\mathbf{x}_t, t)$ , that maps any noisy  $\mathbf{x}_t$  towards the original data. The result is a powerful generative process: sampling  $\mathbf{x}_T \sim N(0, I)$  and applying the learned reverse steps yields new samples  $\mathbf{x}_0$  from the data.

# Denoising Diffusion

- A DM is a generative model consisting of a learned reverse (denoising) process. To sample  $\mathbf{x}_0$  by injecting Gaussian noise, a neural network learns to invert the process.
- Reversible generative process: For



$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = N(\mathbf{x}_t; \sqrt{\alpha_t} \mathbf{x}_{t-1}, (1 - \alpha_t) I)$$

where  $\alpha_t$  is a variance schedule. The learned reverse (denoising) chain is parameterized by  $\theta$ :

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = N(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2 I)$$

- By training the model to predict the added noise, one learns the mean of the Gaussian transition,  $\mu_\theta(\mathbf{x}_t, t)$ , that maps any noisy  $\mathbf{x}_t$  towards the original data. The result is a powerful generative process: sampling  $\mathbf{x}_T \sim N(0, I)$  and applying the learned reverse steps yields new samples  $\mathbf{x}_0$  from the data.

# Diffusion for Unfolding: conditioning

- To apply diffusion to unfolding, we use a conditional diffusion model. Here the reverse process is conditioned on the vertex level  $y$ . Concretely:

$$p_{\theta}(x_{t-1}|x_t, y) = N(x_{t-1}; \mu_{\theta}(x_t, y, t), \sigma_t^2 I)$$

- The forward noising process remains unchanged (just adding noise to  $x$ ), but **the denoiser network now takes  $y$  as an input or context**
- In practice, we train the DM so that after  $T$  steps of reverse diffusion, the output  $x_0$  is a sample from  $P(x|y)$ . Equivalently, the model learns to predict the detector level  $x$  given the ground truth  $y$ . Conditioning directly on  $y$  makes the generative process data-dependent and enables it to approximate the Bayesian inverse  $P(x|y)$  without explicitly computing the prior. Because the learned conditional distribution implicitly encodes the prior, sampling only requires the vertex data.

# Training the Conditional DM

- We generate a large simulated dataset of vertex–detector data  $\{x_0(i), y(i)\}$ , where  $y$  is the vertex level, and  $x_0$  is its detector-level version (smeared by  $g_{\text{sim}}$ ). Each  $x_0(i)$  is then noised to  $x_t$  by adding Gaussian noise according to the variance schedule. The pair  $(x_t, y)$  is used to train the model.
- At each timestep  $t$ , the model  $\epsilon_{\theta}(x_t, y, t)$  predicts the noise that was added. The training loss is the mean-squared error between the true noise  $\epsilon$  and the prediction:

$$L(\theta) = \mathbb{E}_{\mathbf{x}_0, y, \epsilon, t} \left[ \|\epsilon - \epsilon_{\theta}(\mathbf{x}_t, y, t)\|^2 \right]$$

- Minimizing  $L$  encourages the network to accurately reverse the noising process. In effect, the model learns to recover the detector data  $x_0$  from the noisy  $x_t$  given the vertex data  $y$ .

# Pseudocode

- Start with a batch of detector-vertex pairs  $(x_0, y)$ .
- Sample a random  $t$  and noise  $\epsilon \sim N(0, I)$ .
- Compute

$$\mathbf{x}_t = \sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1 - \alpha_t} \epsilon$$

- The network predicts  $e_{\theta}(x_t, y, t)$ .
- Compute  $L$  and backpropagate to update  $\theta$ .
- Iterate until convergence.

# Our aim: a proof of principle for CLAS detector

- Condition (input): vertex information, i.e., the simulated events (we do know the ground truth in this case) in the acceptance of the detector.
- Target: corresponding detector measurements for those same particles (gsim CLAS proxy).

In doing this **we learn how the vertex information gets “smeared” by the detector, i.e., build a function capable of generating realistic detector data from the initial particle properties.**

Training Procedure:

- For each event, take the detector measurement.
- Gradually add noise to that detector data to create increasingly “distorted” versions.
- Train the network to remove that noise, step by step, using the vertex information as context.
- **Actually, the network learns to invert the distortion process: given a noisy version of the detector data, it must recover the clean measurement, knowing the original vertex values.**
- Result of Training: by the end, the model knows how to reconstruct the detector data distribution that corresponds to any given vertex input.

# Our aim: a proof of principle for CLAS detector

- Start from a completely random set of values (pure noise), without any physical information.
- At each step, the network applies its denoising function, guided by the vertex data (which remains the fixed condition).
- After several iterations, the noise is progressively removed to produce a final sample: a set of measurements that mimics exactly what the detector would record for those vertex properties.

## Operational Direction

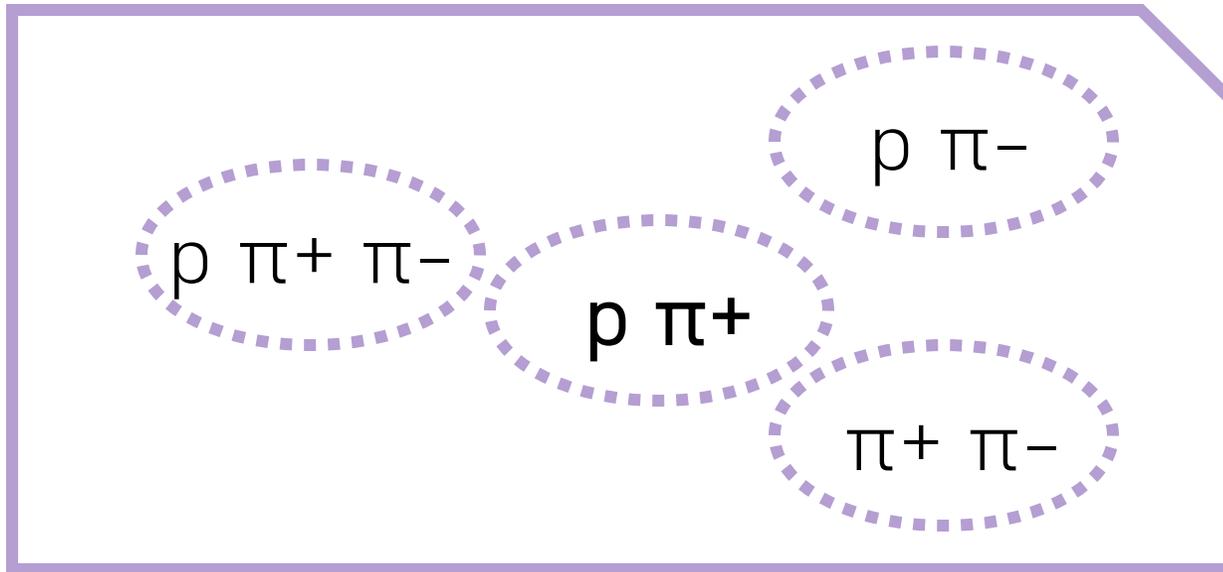
The flow goes from pure noise, through many “cleaning” steps, to produce synthetic detector data. In this way, the network has learned to emulate the detector’s response.

## Validating Results:

- Compare the characteristics (e.g., histogram shapes up to now) of the generated data with those of detector data.
- If the distributions match, it means the model has correctly learned how the detector transforms the vertex information.

# Our strategy

- Let's start from the most populated topology ( $\rho \pi^+$ )



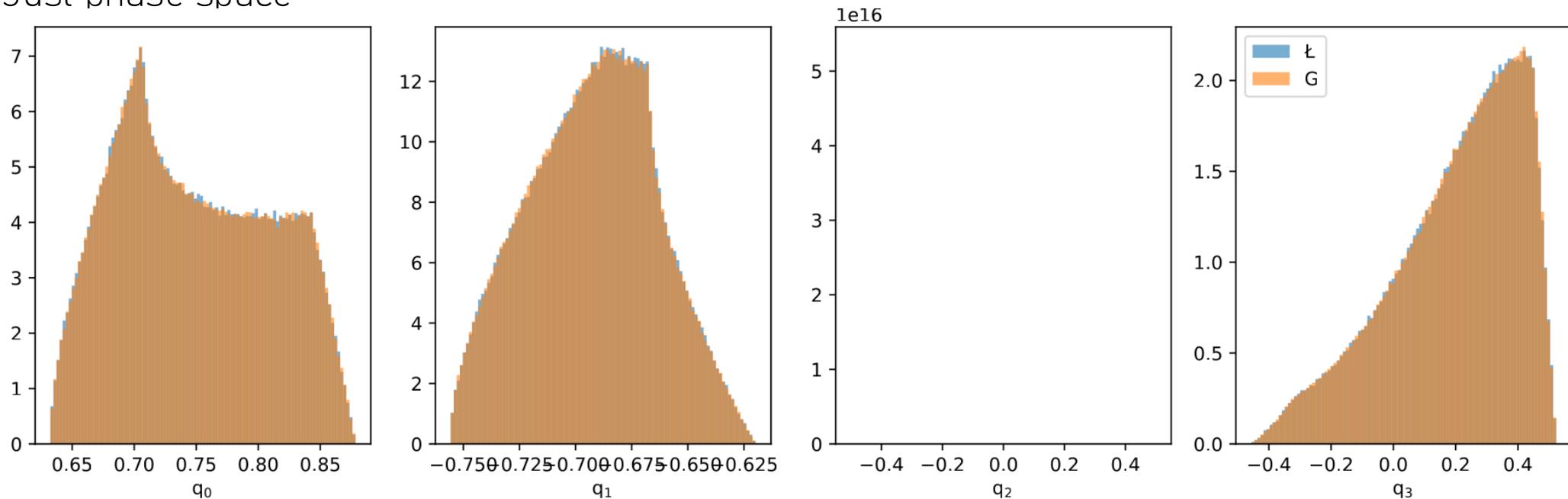
Phase space region covered by CLAS

- Then we will exploit DMs flexibility for moving from one topology to the other

# Our strategy

Using MC generated 4-vectors sampled from different models and correspondent gsim reconstructed and train the DM on them:

- Just phase space



- More refined model with resonances included at the level of cross sections
- Regge-inspired model (by JPAC collaboration) with resonances included at the level of the amplitude

Let's define the moments

$$\frac{d\sigma}{dt d\sqrt{s_{12}} d\Omega^H} = \kappa \sum_{\lambda_1 \lambda_\gamma \lambda_2} |\mathcal{M}_{\lambda_\gamma \lambda_1 \lambda_2}(s, t, s_{12}, \Omega^H)|^2,$$

$$\langle Y_M^L \rangle = \sqrt{4\pi} \int d\Omega^H \frac{d\sigma}{dt d\sqrt{s_{12}} d\Omega^H} \text{Re}\{Y_M^L(\Omega^H)\}$$

$$\mathcal{M}_{\lambda_\gamma \lambda_1 \lambda_2}(s, t, s_{12}, \Omega^H) = \sum_{lm} \mathcal{M}_{\lambda_\gamma \lambda_1 \lambda_2 m}^l(s, t, s_{12}) Y_m^l(\Omega^H),$$

$$\begin{aligned} \langle Y_M^L \rangle &= \sqrt{4\pi} \kappa \sum_{lm'l'm'} A_{Mmm'}^{Ll'l'} \\ &\times \sum_{\lambda_\gamma \lambda_1 \lambda_2} \mathcal{M}_{\lambda_\gamma \lambda_1 m' \lambda_2}^{l'*}(s, t, s_{12}) \mathcal{M}_{\lambda_\gamma \lambda_1 m \lambda_2}^l(s, t, s_{12}), \end{aligned}$$

where

$$A_{Mmm'}^{Ll'l'} = \int d\Omega^H Y_m^l(\Omega^H) Y_{m'}^{l'*}(\Omega^H) \text{Re}\{Y_M^L(\Omega^H)\}.$$

# Let's define the moments

Photoproduction of  $\pi^+\pi^-$  meson pairs on the proton

Moment extraction  
from real data by  
CLAS collaboration

$\frac{d\sigma}{d\Omega d\Omega'} d\Omega'$

M. Battaglieri,<sup>1</sup> R. De Vita,<sup>1</sup> A. P. Szczepaniak,<sup>2</sup> K. P. Adhikari,<sup>35</sup> M.J. Amarian,<sup>35</sup> M. Anghinolfi,<sup>1</sup>  
 H. Baghdasaryan,<sup>45</sup> I. Bedlinskiy,<sup>22</sup> M. Bellis,<sup>7</sup> L. Bibrzycki,<sup>29</sup> A.S. Biselli,<sup>13,36</sup> C. Bookwalter,<sup>15</sup>  
 D. Branford,<sup>12</sup> W.J. Briscoe,<sup>16</sup> V.D. Burkert,<sup>42</sup> S.L. Careccia,<sup>35</sup> D.S. Carman,<sup>42</sup> E. Clinton,<sup>28</sup> P.L. Cole,<sup>18</sup>  
 P. Collins,<sup>4</sup> V. Crede,<sup>15</sup> D. Dale,<sup>18</sup> A. D'Angelo,<sup>20,38</sup> A. Daniel,<sup>34</sup> N. Dashyan,<sup>47</sup> E. De Sanctis,<sup>19</sup> A. Deur,<sup>42</sup>  
 S. Dhamija,<sup>14</sup> C. Djalali,<sup>41</sup> G.E. Dodge,<sup>35</sup> D. Doughty,<sup>10,42</sup> V. Drozdov,<sup>1</sup> H. Egiyan,<sup>30,42</sup> P. Eugenio,<sup>15</sup>  
 G. Fedotov,<sup>40</sup> S. Fegan,<sup>17</sup> A. Fradi,<sup>21</sup> M.Y. Gabrielyan,<sup>14</sup> L. Gan,<sup>32</sup> M. Garçon,<sup>9</sup> A. Gasparian,<sup>33</sup> G.P. Gilfoyle,<sup>37</sup>  
 K.L. Giovanetti,<sup>24</sup> F.X. Girod,<sup>9,\*</sup> O. Glamazdin,<sup>26</sup> J. Goett,<sup>36</sup> J.T. Goetz,<sup>5</sup> W. Gohn,<sup>11</sup> E. Golovatch,<sup>40,1</sup>  
 R.W. Gothe,<sup>41</sup> K.A. Griffioen,<sup>46</sup> M. Guidal,<sup>21</sup> L. Guo,<sup>42,†</sup> K. Hafidi,<sup>3</sup> H. Hakobyan,<sup>44,47</sup> C. Hanretty,<sup>15</sup>  
 N. Hassall,<sup>17</sup> K. Hicks,<sup>34</sup> M. Holtrop,<sup>30</sup> C.E. Hyde,<sup>35</sup> Y. Ilieva,<sup>41,16</sup> D.G. Ireland,<sup>17</sup> E.L. Isupov,<sup>40</sup> J.R. Johnstone,<sup>17</sup>  
 K. Joo,<sup>11</sup> D. Keller,<sup>34</sup> M. Khandaker,<sup>31</sup> P. Khetarpal,<sup>36</sup> W. Kim,<sup>27</sup> A. Klein,<sup>35</sup> F.J. Klein,<sup>8</sup> M. Kossov,<sup>22</sup>  
 A. Kubarovsky,<sup>35</sup> V. Kubarovsky,<sup>42</sup> S.V. Kuleshov,<sup>44,22</sup> V. Kuznetsov,<sup>27</sup> J.M. Laget,<sup>42,9</sup> L. Lesniak,<sup>29</sup>  
 K. Livingston,<sup>17</sup> H.Y. Lu,<sup>41</sup> M. Mayer,<sup>35</sup> M.E. McCracken,<sup>7</sup> B. McKinnon,<sup>17</sup> C.A. Meyer,<sup>7</sup> K. Mikhailov,<sup>22</sup>  
 T. Mineeva,<sup>11</sup> M. Mirazita,<sup>19</sup> V. Mochalov,<sup>23</sup> V. Mokeev,<sup>40,42</sup> K. Moriya,<sup>7</sup> E. Munevar,<sup>16</sup> P. Nadel-Turonski,<sup>8</sup>  
 I. Nakagawa,<sup>39</sup> C.S. Nepali,<sup>35</sup> S. Niccolai,<sup>21</sup> I. Niculescu,<sup>24</sup> M.R. Niroula,<sup>35</sup> M. Osipenko,<sup>1,40</sup> A.I. Ostrovidov,<sup>15</sup>  
 K. Park,<sup>41,27,\*</sup> S. Park,<sup>15</sup> M. Paris,<sup>16,42</sup> E. Pasyuk,<sup>4</sup> S. Anefalos Pereira,<sup>19</sup> S. Pisano,<sup>21</sup> N. Pivnyuk,<sup>22</sup>  
 O. Pogorelko,<sup>22</sup> S. Pozdniakov,<sup>22</sup> J.W. Price,<sup>6</sup> Y. Prok,<sup>45,‡</sup> D. Protopopescu,<sup>17</sup> B.A. Raue,<sup>14,42</sup> G. Ricco,<sup>1</sup>  
 M. Ripani,<sup>1</sup> B.G. Ritchie,<sup>4</sup> G. Rosner,<sup>17</sup> P. Rossi,<sup>19</sup> F. Sabatié,<sup>9</sup> M.S. Saini,<sup>15</sup> C. Salgado,<sup>31</sup> D. Schott,<sup>14</sup>  
 R.A. Schumacher,<sup>7</sup> H. Seraydaryan,<sup>35</sup> Y.G. Sharabian,<sup>42</sup> D.I. Sober,<sup>8</sup> D. Sokhan,<sup>12</sup> A. Stavinsky,<sup>22</sup> S. Stepanyan,<sup>42</sup>  
 S. S. Stepanyan,<sup>27</sup> P. Stoler,<sup>36</sup> I.I. Strakovsky,<sup>16</sup> S. Strauch,<sup>41,16</sup> M. Taiuti,<sup>1</sup> D.J. Tedeschi,<sup>41</sup> A. Teymurazyan,<sup>25</sup>  
 S. Tkachenko,<sup>35</sup> M. Ungaro,<sup>11,36</sup> M.F. Vineyard,<sup>43</sup> A.V. Vlassov,<sup>22</sup> D.P. Watts,<sup>17,§</sup> L.B. Weinstein,<sup>35</sup>  
 D.P. Weygand,<sup>42</sup> M. Williams,<sup>7</sup> E. Wolin,<sup>42</sup> M.H. Wood,<sup>41</sup> L. Zana,<sup>30</sup> J. Zhang,<sup>35</sup> B. Zhao,<sup>11,¶</sup> and Z.W. Zhao<sup>41</sup>

(The CLAS Collaboration)

$$A_{Mmm'}^{Ll'l'} = \int d\Omega^H Y_m^l(\Omega^H) Y_{m'}^{l'*}(\Omega^H) \text{Re}\{Y_M^L(\Omega^H)\}.$$

$$\frac{d\sigma}{2d\Omega^H} \text{Re}\{Y_M^L(\Omega^H)\}$$

# Fit results by JPAC collaboration

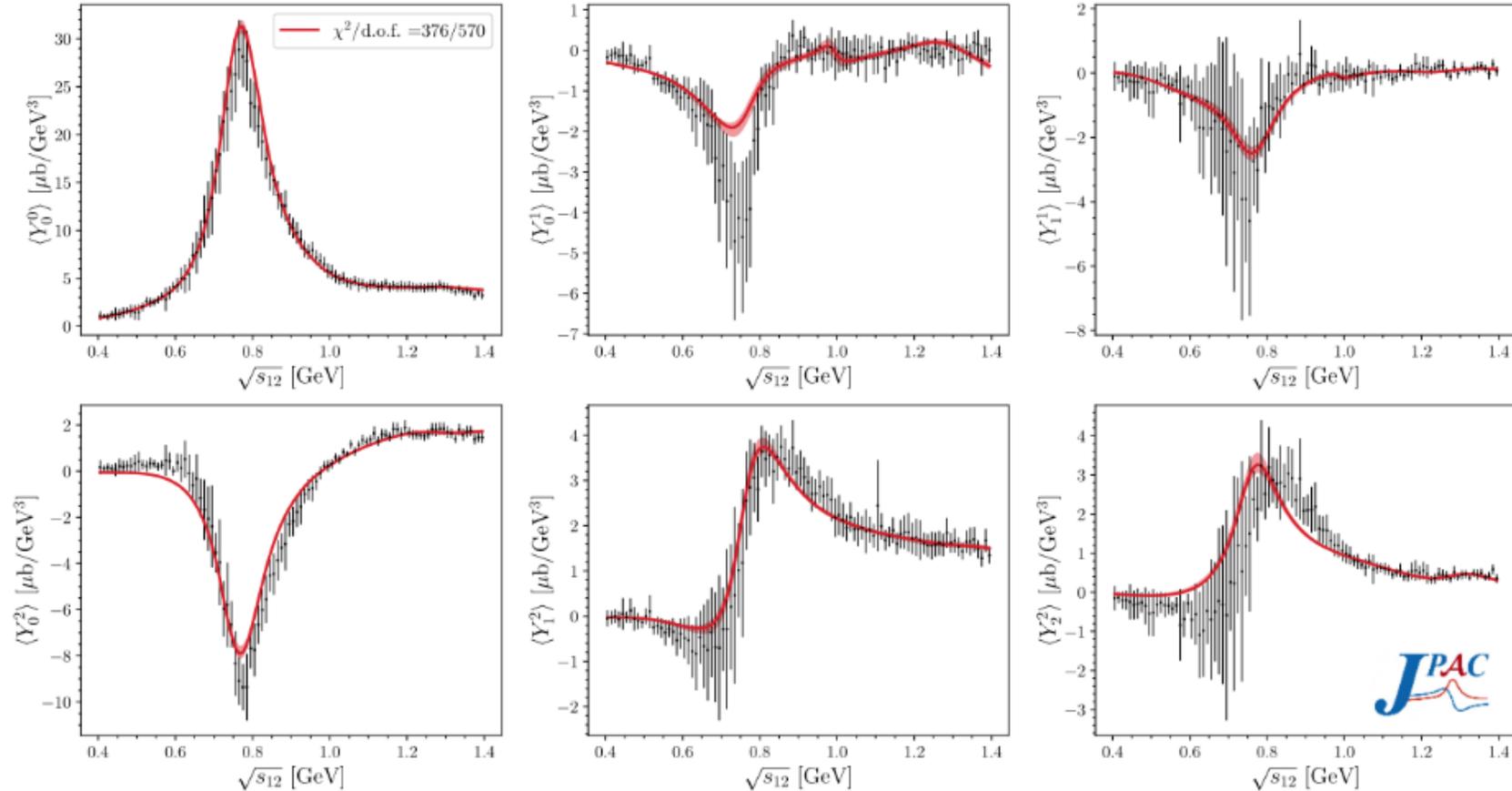


FIG. 5. Comparison of complete model fitted to experimental measurements from Ref. [21] of two-pion angular moments  $\langle Y_M^L \rangle$  for  $L = 0, 1, 2$  and  $M = 0, \dots, L$  for  $E_\gamma = 3.7$  GeV and  $t = -0.45$  GeV<sup>2</sup>. Since all data shown here are fit simultaneously, this corresponds to  $600 - 30 = 570$  degrees of freedom (d.o.f.).

## Studying $\pi^+\pi^-$ photoproduction beyond Pomeron exchange

Łukasz Bibrzycki<sup>1,\*</sup>, Nadine Hammoud<sup>2,3,†</sup>, Vincent Mathieu<sup>3</sup>, Robert J. Perry<sup>3,‡</sup>, Alex Akridge<sup>4,5</sup>,  
 César Fernández-Ramírez<sup>6</sup>, Gloria Montaña<sup>7</sup>, Alessandro Pilloni<sup>8,9</sup>, Arkaitz Rodas<sup>7,10</sup>, Vanamali Shastry<sup>4,5</sup>,  
 Wyatt A. Smith<sup>11,12,13</sup>, Daniel Winney<sup>14</sup> and Adam P. Szczepaniak<sup>7,5,4</sup>

# Regge model used to fit the experimental moments

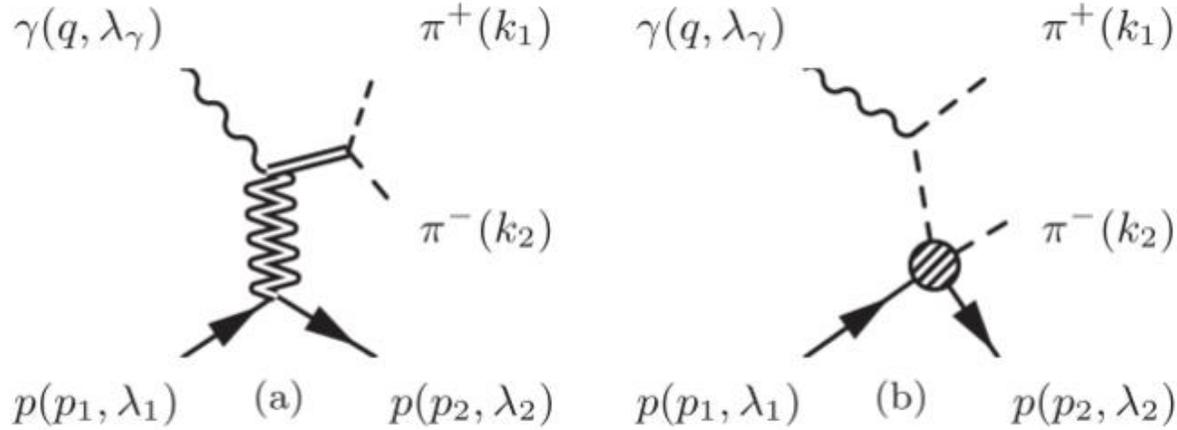


FIG. 4. The scattering amplitude for two pion photoproduction can be written as a sum of terms as shown in Eq. (12). Dominant contributions at small momentum transfer can be separated into two categories based on their Feynman diagram topology. Diagram (a) shows the production of resonance  $\mathcal{R}$  (straight double line) via the exchange of a Reggeon,  $\mathbb{R}$  (zig-zag double line). Diagram (b) represents the Deck contribution. Note that there exists another diagram with the topology of (b) where the charged pions are exchanged. With this approximation, it is possible to relate this  $2 \rightarrow 3$  process to  $2 \rightarrow 2$  processes.

TABLE I. Resonance parameters employed in this work. Values are taken from Ref. [23].

$\mathcal{R}$	$J$	$m_{\mathcal{R}}$ [GeV]	$\Gamma_{\mathcal{R}}$ [GeV]
$f_0(500)$	0	0.500	0.450
$\rho(770)$	1	0.775	0.149
$f_0(980)$	0	0.990	0.055
$f_2(1270)$	2	1.2755	0.1867
$f_0(1370)$	0	1.370	0.350

$$a_M^{\mathbb{P},\rho}(t) = g_{\gamma\mathbb{P}\rho} g_{\rho\pi\pi} \beta_{\mathbb{P}}(t), \quad M = -1, 0, 1 \quad (22)$$

are fixed with  $g_{\gamma\mathbb{P}\rho} = 5.96$ ,  $g_{\rho\pi\pi} = 2.506$  and  $\beta_{\mathbb{P}}(t) = \exp(bt)$  with  $b = 3.6 \text{ GeV}^{-2}$  are taken from Ref. [19], while the parameters

$$a^{\rho/\omega, f_0(500)}(t), \quad a^{\rho/\omega, f_0(980)}(t), \quad a^{\rho/\omega, f_0(1370)}(t), \\ a_M^{a_2/f_2, \rho}(t), \quad a_M^{\rho/\omega, f_2}(t), \quad (23)$$

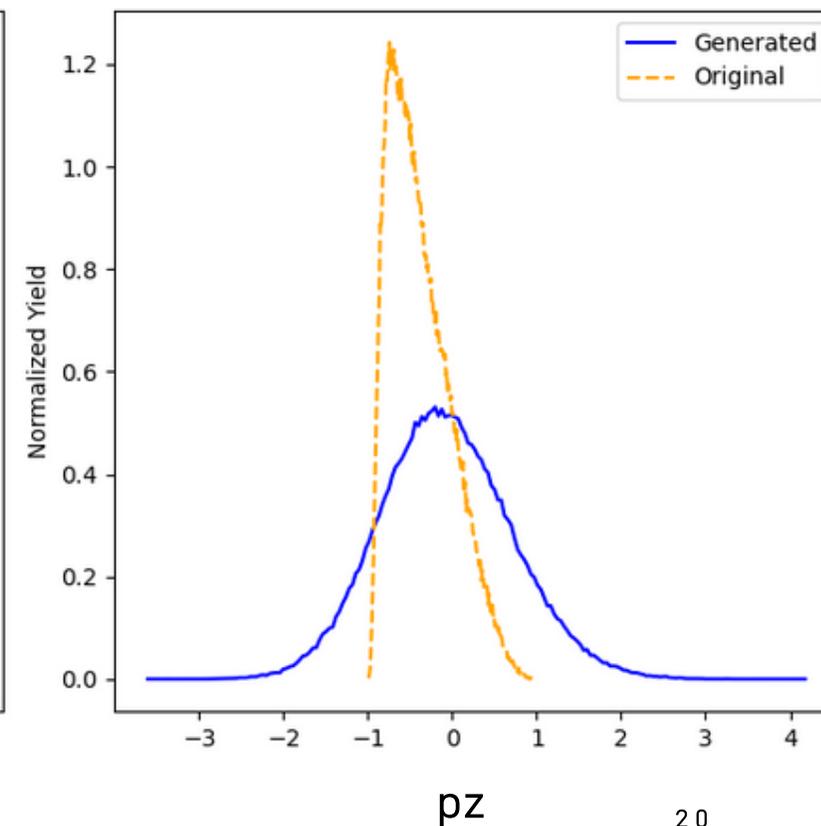
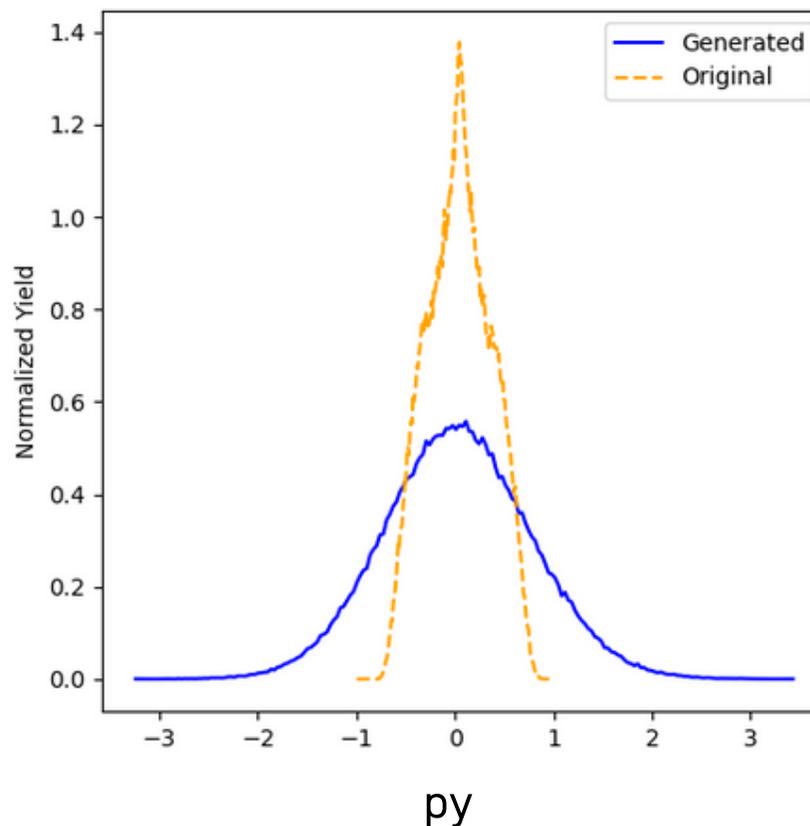
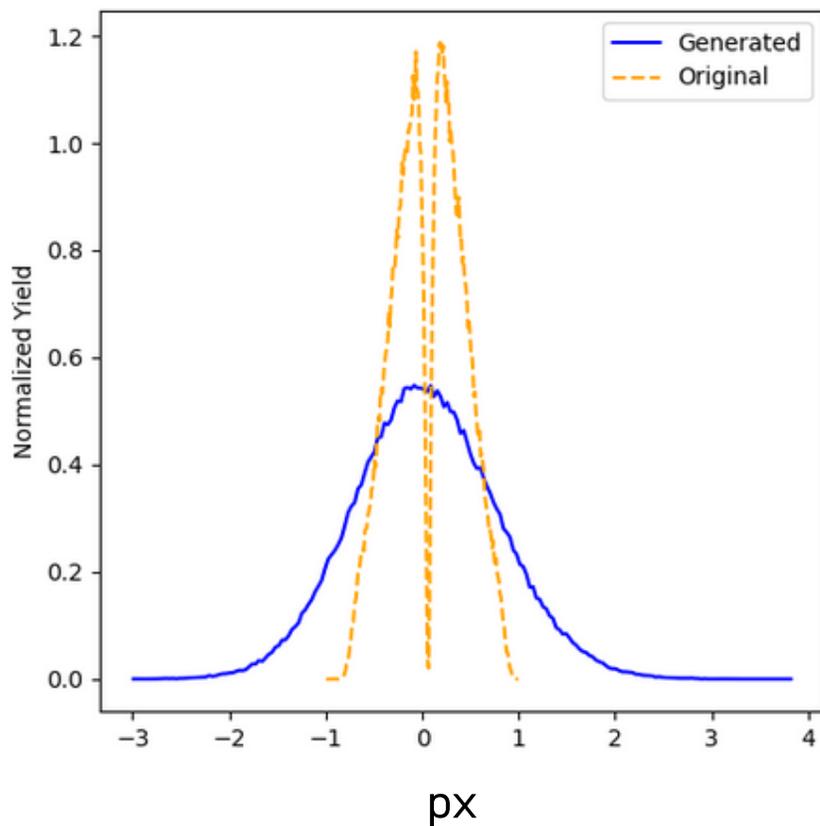
# Current state of the project:

- We are training the diffusion model on simulations based on phase space only: generated in the acceptance + reconstructed (gsim detector proxy)

(proton 3-momentum)

Epoch 5/10000 [Loss: 0.209704]

Sampling: 100% | ██████████ | 100/100 [00:34<00:00, 2.92it/s]



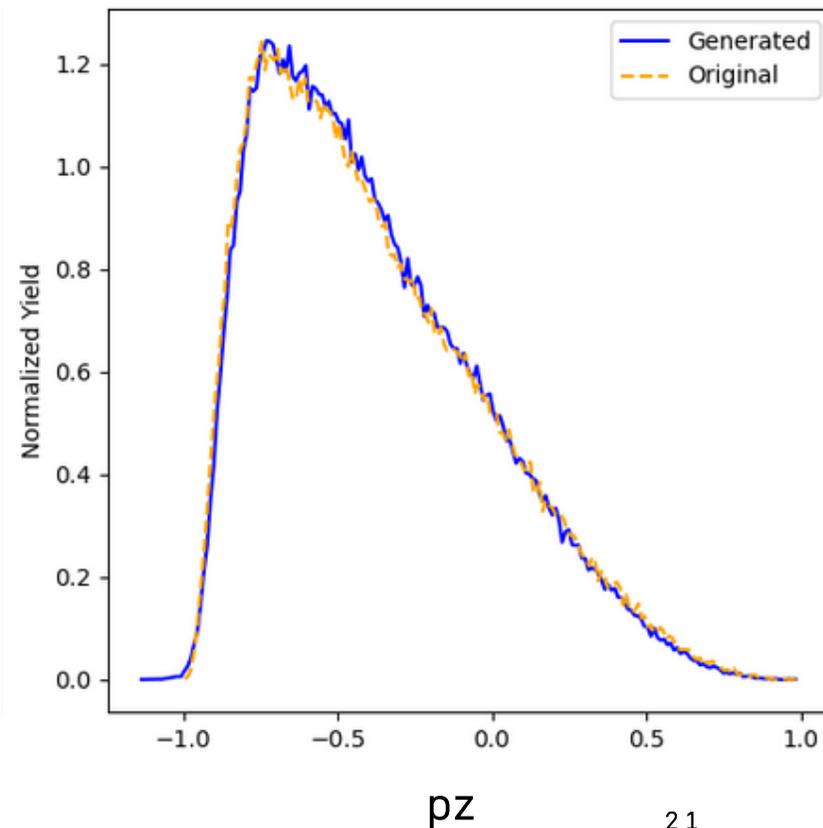
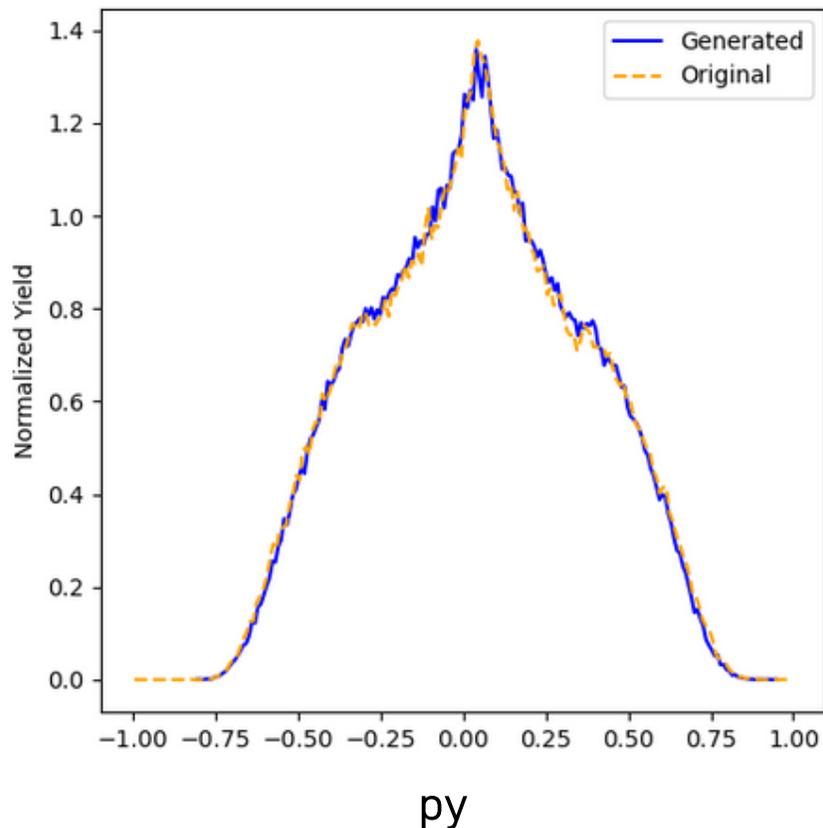
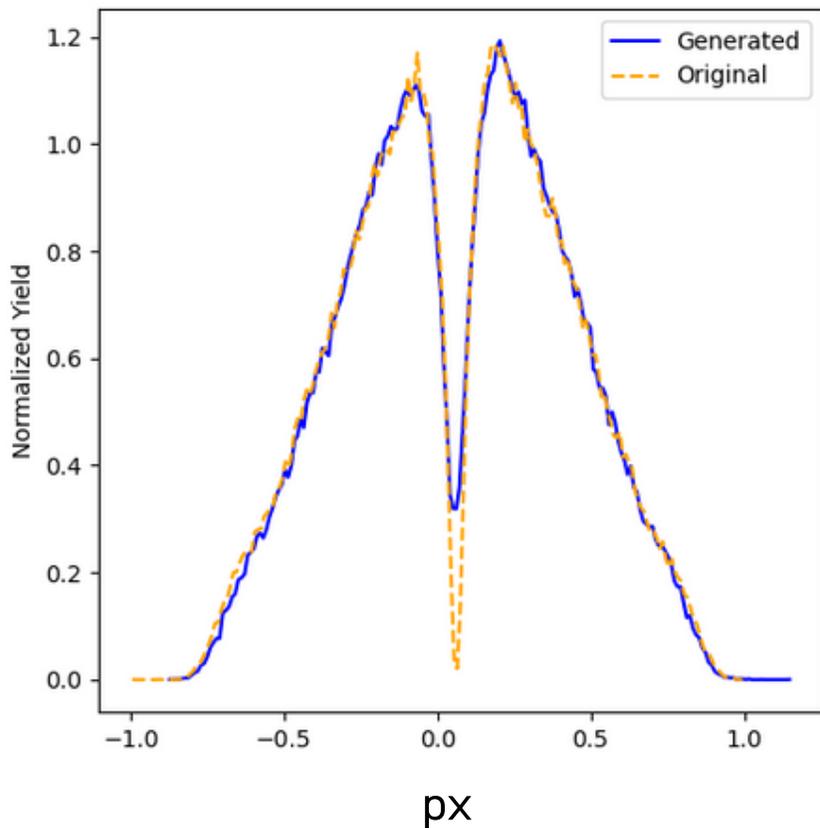
# Current state of the project:

- We are training the diffusion model on simulations based on phase space only: generated in the acceptance + reconstructed (gsim detector proxy)

Epoch 95/10000 [Loss: 0.018124]

Sampling: 100% | 100/100 [00:33<00:00, 2.95it/s]

(proton 3-momentum)



# Next steps

- **Uncertainty quantification:** While diffusion models are inherently stochastic, rigorous uncertainty estimates for unfolded results are needed. Approaches include Bayesian treatments of the model or ensembles. Studies note that incorporating **uncertainty estimation** (e.g. sampling variance, systematic errors) is crucial for practical use.
- **Domain adaptation:** Real data may not perfectly match the simulation used for training. Domain-adaptation techniques (e.g. adversarial learning, transfer learning) could adapt the cDM to detector-data differences without labels.
- **Handling edge cases:** Special attention must be given to particles outside detector acceptance or rare phase-space regions. The model should ideally be able to flag or properly treat such outliers.
- **Systematics and physics constraints:** Integrating known systematic uncertainties and enforcing physical constraints (like energy-momentum conservation) during sampling remain open challenges.

---

# Future directions

- Enlarge as much as possible the measured phase space adding **all topologies** through this diffusion model procedure
- **Application of our DM & Collaboration with GlueX** people to enlarge it way more including the phase space measured by them: note that – independently on the techniques that they will use to unfold GlueX detector effect – we will be able to compare momenta distribution extracted at the vertex level by us and them in the region overlapped by the two detectors -> this is another sanity check that our DM-unfolding procedure works
- This – as far as we know – never tried approach can reduce a lot the model dependence that we introduce extrapolating the cross section in the unmeasured phase space region