Towards Universal Unfolding using Denoising Diffusion Probabilistic Models

Camila Pazos, Shuchin Aeron, Pierre-Hugues Beauchemin,
Vincent Croft, Martin Klassen, Taritree Wongjirad

Presented by: Camila Pazos

France-Berkeley PHYSTAT Conference on Unfolding
13 June 2024

Tufts UNIVERSITY

Universiteit Leiden The Netherlands

U.S. DEPARTMENT OF ENERGY

NSF

Outline

→ **Results!**

I. The Unfolding Problem

II. Denoising Diffusion Probabilistic Models

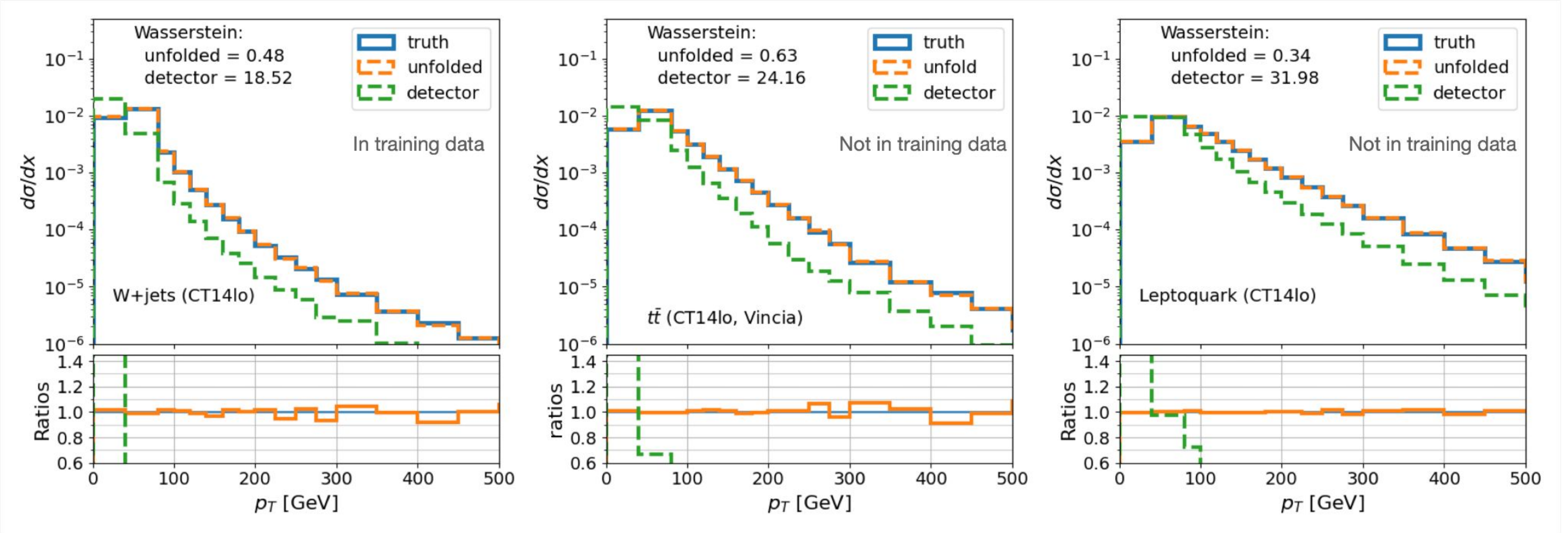
III. cDDPM Unfolder

IV. Our Approach

V. Summary

Results

Using a single conditional Denoising Diffusion Probabilistic Model, we are able to perform multidimensional object-wise unfolding of detector data from various physics processes, including those not seen in the training data.



The Unfolding Problem

- Detector distortions affect the kinematic quantities of particles incident to the detector.
- Use statistical tools to infer the true underlying distribution $f_{\text{true}}(x)$ from the observed distribution $f_{\text{data}}(y)$ obtained from experiment data.

- For detector effects $P(y\,|\,x)$,

$$f_{\text{data}}(y) = \int dx\, P(y\,|\,x) f_{\text{true}}(x)$$

- To unfold, we can estimate the inverse process $P(x\,|\,y)$,

$$P(x\,|\,y) = \frac{P(y\,|\,x)\, f_{\text{true}}(x)}{f_{\text{data}}(y)}$$

→ A model that learns the posterior $P(x\,|\,y)$ will be consistent with the selected prior $f_{\text{true}}(x)$.

Denoising Diffusion Probabilistic Models

→ Denoising Diffusion Probabilistic Models

→ Conditional DDPM

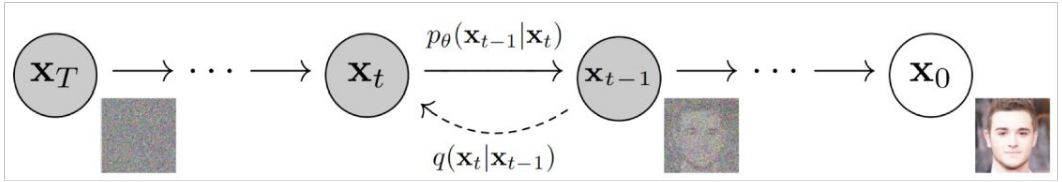
Denoising Diffusion Probabilistic Models

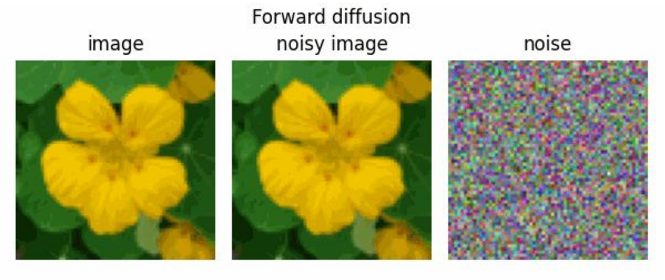
Forward (Diffusion) Process:
Adds noise in steps to a dataset following a variance schedule $\beta_1, \dots, \beta_T$

$$q(x_t \,|\, x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}\, x_t, \beta_t \mathbf{I})$$

Reverse (Denoising) Process:
The model learns to reverse the diffusion process, denoising the dataset over T steps to recover the original data distribution.

$$p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \,|\, x_t)$$


$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$\mathbf{x}_0$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$



Denoising Diffusion Probabilistic Models

Forward (Diffusion) Process:

Adds noise in steps to a dataset following a variance schedule $\beta_1, \dots, \beta_T$

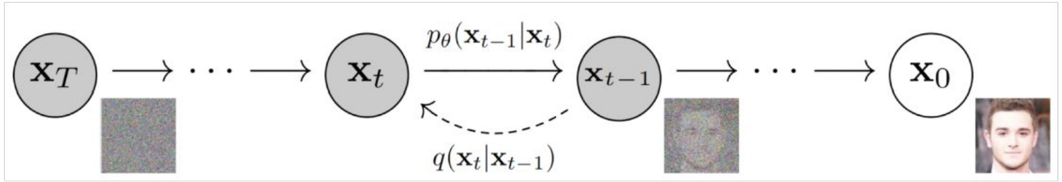
$$q(x_t \mid x_{t-1}) := \mathcal{N}(x_t ; \sqrt{1-\beta_t}\, x_t, \beta_t \mathbf{I})$$

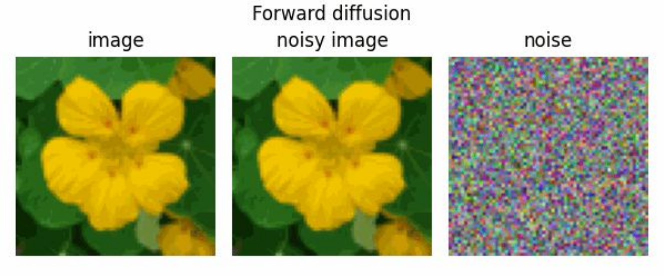
To use in unfolding, we want to condition the model on our detector data y

Reverse (Denoising) Process:

The model learns to reverse the diffusion process, denoising the dataset over T steps to recover the original data distribution.

$$p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$$


$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$\mathbf{x}_0$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$



Conditional Denoising Diffusion Probabilistic Models

Forward (Diffusion) Process:

Adds noise in steps to a dataset following a variance schedule $\beta_1, \dots, \beta_T$

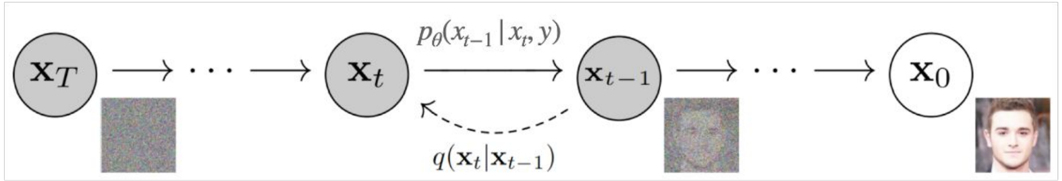
$$q(x_t | x_{t-1}) := \mathcal{N}(x_t ; \sqrt{1-\beta_t}\, x_t, \beta_t \mathbf{I})$$

Keep the forward process the same

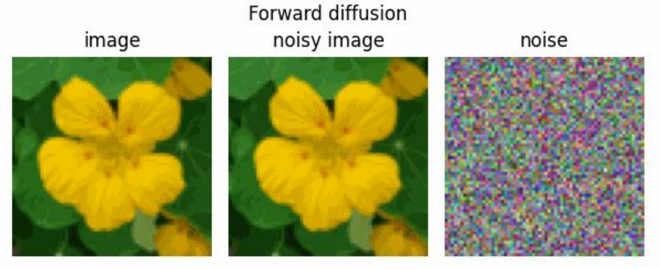
Reverse (Denoising) Process:

The model learns to reverse the diffusion process conditioned on an input y, denoising the dataset over T steps to recover the original data distribution.

$$\cancel{p_\theta(x_{0:T}) := p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1} | x_t)} \xrightarrow{\text{Replace}} p_\theta(x_{0:T} | y) := p(x_T | y)\prod_{t=1}^{T} p_\theta(x_{t-1} | x_t, y)$$



$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(x_{t-1} | x_t, y)$
$\mathbf{x}_{t-1}$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_0$



cDDPM Unfolder

→ cDDPM Unfolder Setup

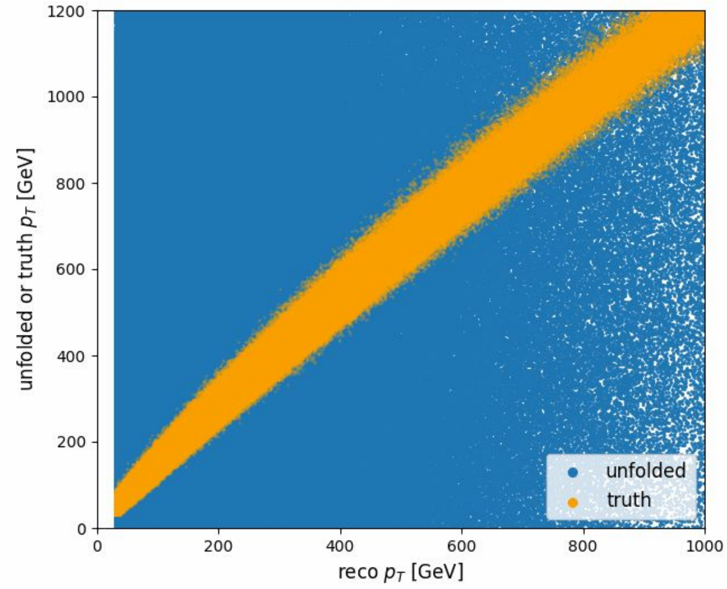
→ Physics Results

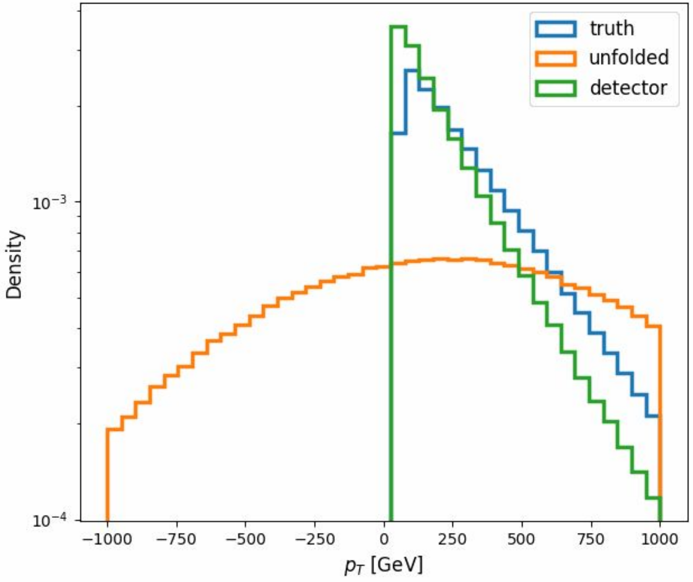
→ Dependence on Training Prior

→ Generalization?

cDDPM Unfolder Setup

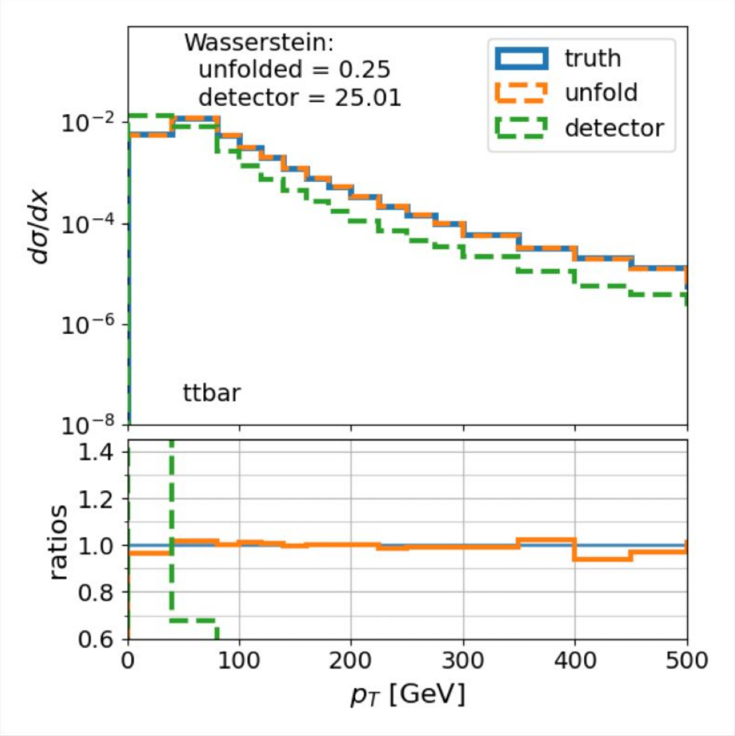
- Each object is defined with a vector $[p_T, \eta, \phi, E, p_x, p_y, p_z]$ at truth-level $(\vec{x})$ and detector-level $(\vec{y})$.
- Train a cDDPM using dataset pairs $\{\vec{x}, \vec{y}\}$ to learn to sample from the posterior $P(\vec{x} \mid \vec{y})$.
- Given detector data $\vec{y}$, we sample from the cDDPM to recover the truth-level $\vec{x}$.

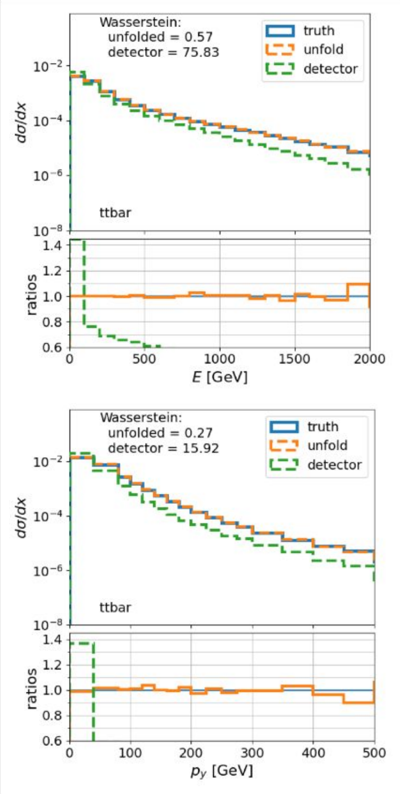


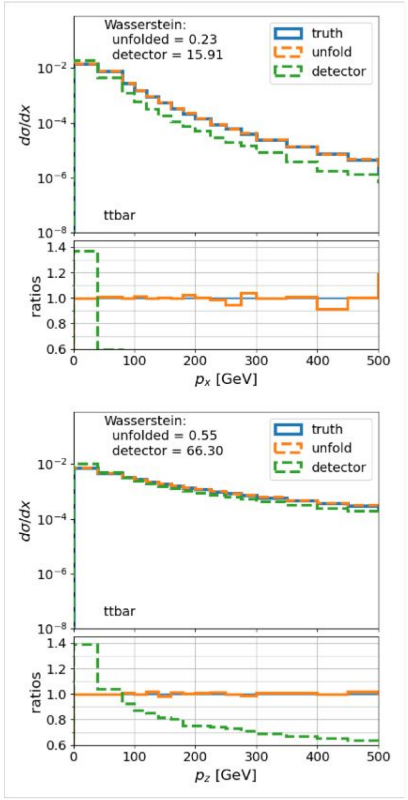


Physics Results

We tested the cDDPM unfolding with simulated $t\bar{t}$ jets data. (Closure test)





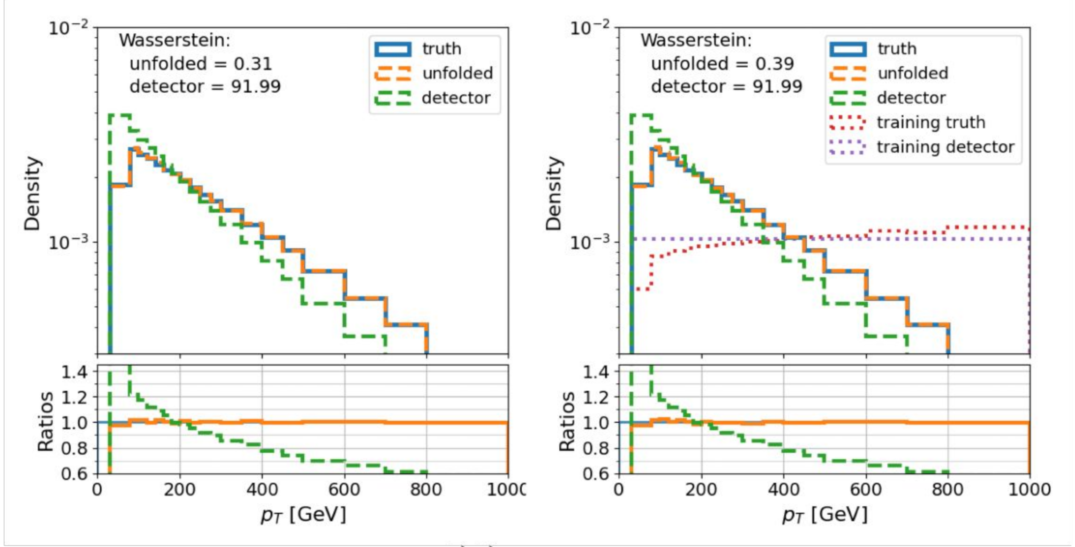


Dependence on Training Prior

The cDDPM formulation learns to sample from the posterior $P(\vec{x}\,|\,\vec{y})$ without explicit use of the training distribution prior.

Using toy data…

1. Define two datasets, denoted with i and j, where
   - Same posteriors:
     $$P_i(\vec{x}\,|\,\vec{y}) = P_j(\vec{x}\,|\,\vec{y})$$
   - Different marginals:
     $$f^i_{\text{true}}(x) \neq f^j_{\text{true}}(x) \text{ and } f^i_{\text{data}}(y) \neq f^j_{\text{data}}(y)$$
2. Train a cDDPM using a dataset $\{\vec{x},\vec{y}\}_j \sim P_j(\vec{x},\vec{y})$
3. Use it to unfold $\{\vec{y}\}_i \sim f^i_{\text{data}}(y)$

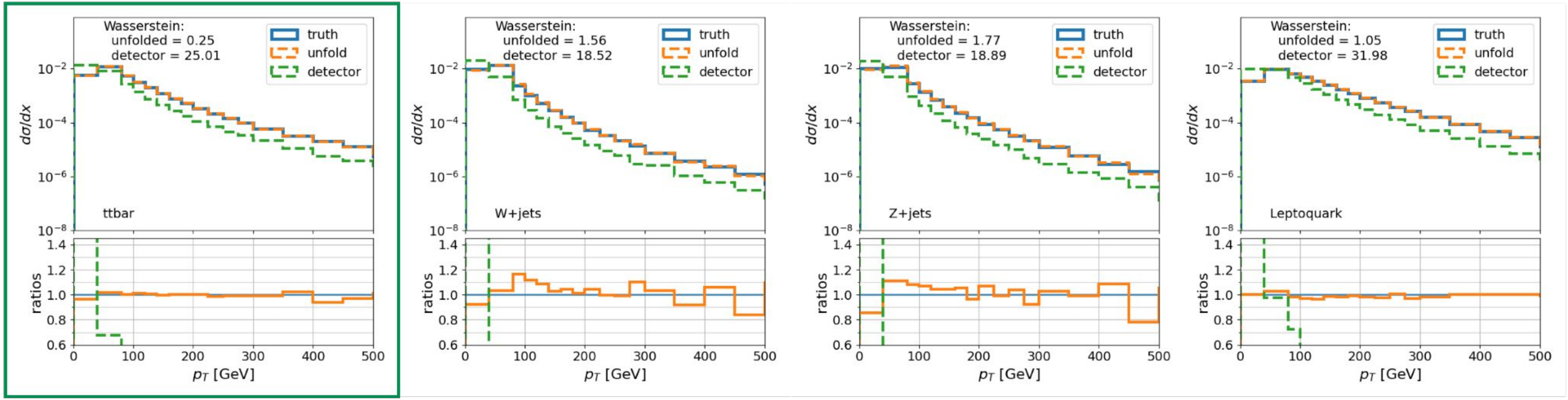


Test *and* training data $\sim P_i(\vec{x},\vec{y})$

Test data $\sim P_i(\vec{x},\vec{y})$, while training data $\sim P_j(\vec{x},\vec{y})$

Generalization?

- How different or similar are the posteriors between different physics processes?
- We tested the generalization by using a cDDPM trained with $t\bar{t}$ jets data to unfold jets from other processes.



Our Approach

→ Generalization and Moments

→ Generalizable cDDPM Unfolder Setup

→ Physics Results

→ Additional Tests

→ Maintaining Correlations

Generalization and Moments

For two different physics processes i and j under the same detector effects,

$$\frac{P_i(x\,|\,y)}{P_j(x\,|\,y)} = \frac{f^i_{\text{true}}(x)}{f^i_{\text{data}}(y)}\,\frac{f^j_{\text{data}}(y)}{f^j_{\text{true}}(x)}$$

→ If we can learn a posterior $P_i(x\,|\,y)$, then we could extrapolate to an unseen posterior $P_j(x\,|\,y)$ by utilizing information about their marginals!

How can we acquire and utilize distributional information of $f_{\text{true}}(x)$ and $f_{\text{data}}(y)$?

→ Calculate the first moments of these distributions and incorporate them into the datasets

→ Include this information in the conditioning and generative aspects of our machine learning model

Generalizable cDDPM Unfolder Setup

→ Use the same cDDPM structure as before, only change the model inputs!

Training Dataset

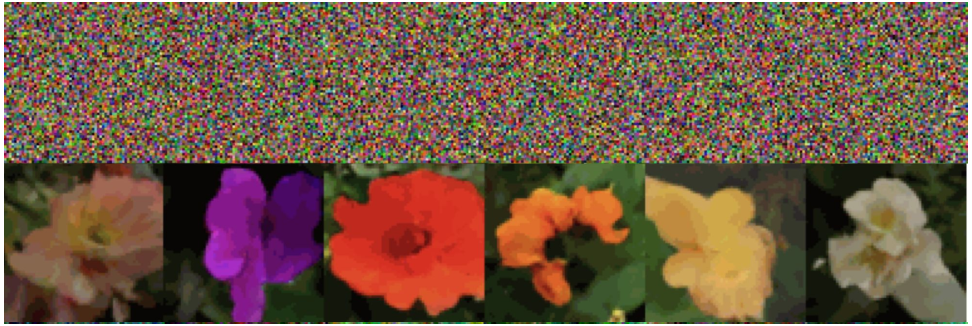
- For various physics processes i, calculate the first 6 moments of the p_T distributions of each dataset $\{\vec{x}, \vec{y}\}_i$
- Append the moments to the data vectors to get moments-included datasets $\{\vec{x}^m, \vec{y}^m\}_i$
- Combine to form the training dataset $\{\vec{x}^m, \vec{y}^m\}_{\text{train}} = \cup_i \{\vec{x}^m, \vec{y}^m\}_i$

Training

- Train a cDDPM using the training dataset $\{\vec{x}^m, \vec{y}^m\}_{\text{train}}$ to learn to sample from the posteriors $P_i(\vec{x}^m \,|\, \vec{y}^m)$

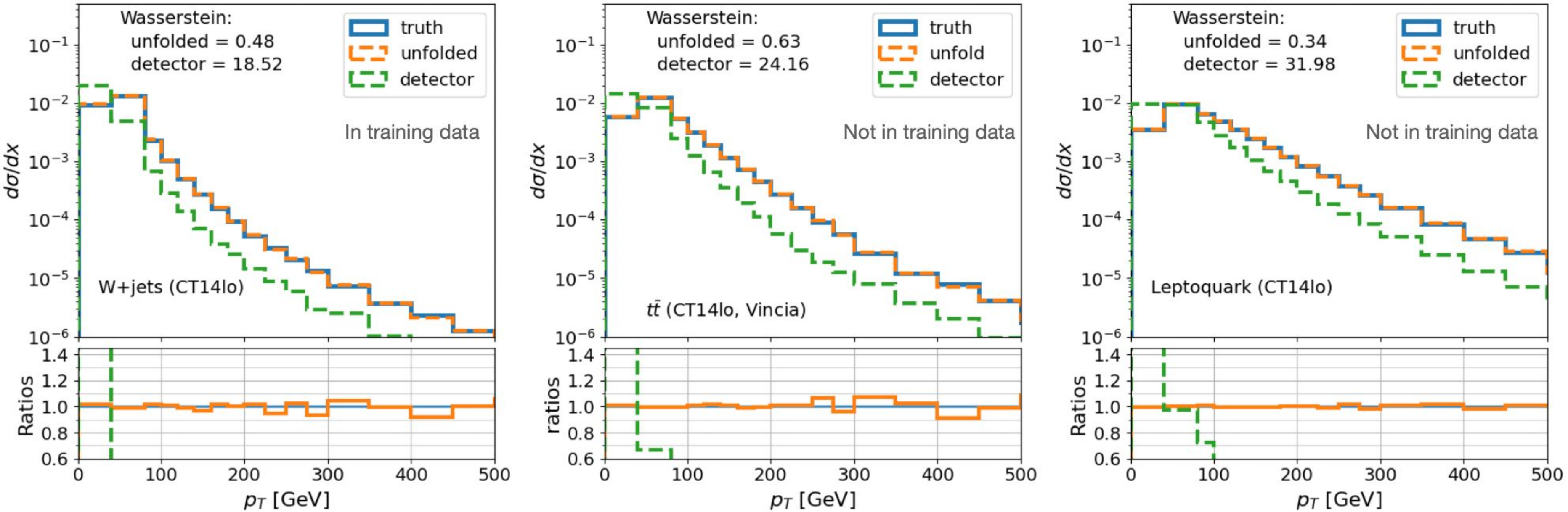
Sampling

- Given a detector dataset $\{\vec{y}^m\}_j$, sample from the estimated posterior $P_j(\vec{x}^m \,|\, \vec{y}^m)$ to recover $\{\vec{x}^m\}_j$



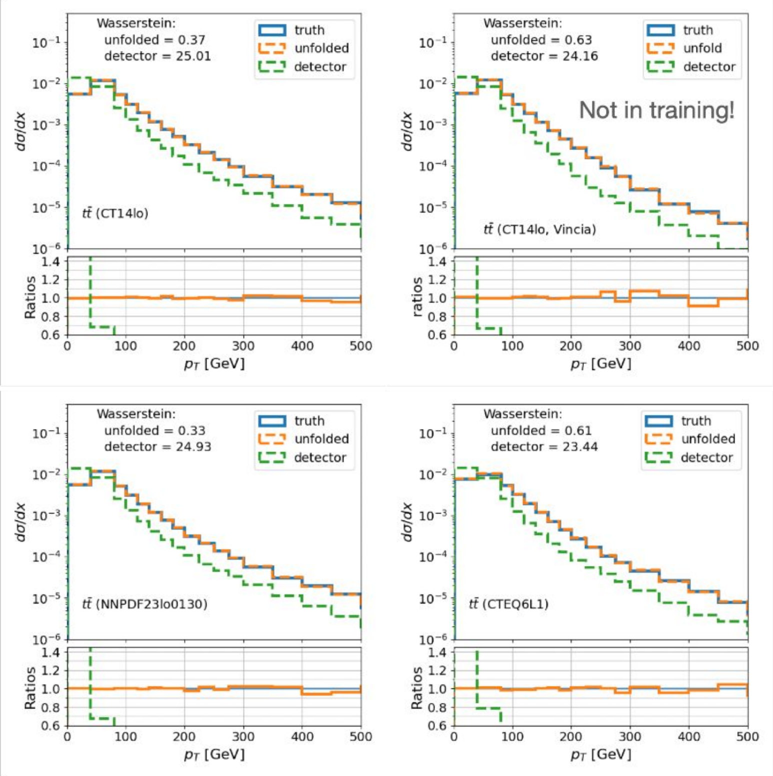
Physics Results

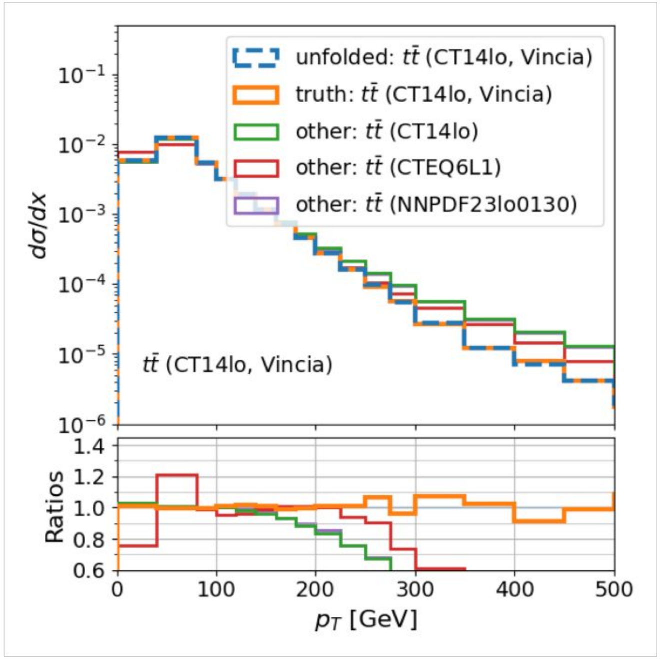
- Including the moments in the conditioning and generative aspects of the cDDPM allows it to learn multiple posteriors $P_i(\vec{x}\,|\,\vec{y})$ and extrapolate to unseen posteriors.
- Using a single cDDPM as a posterior sampler, we can unfold jets data from multiple physics processes, including those not seen during the training.



Physics Results

- We also performed tests by unfolding different versions of $t\bar{t}$ simulations (varying the PDFs and the parton shower model).

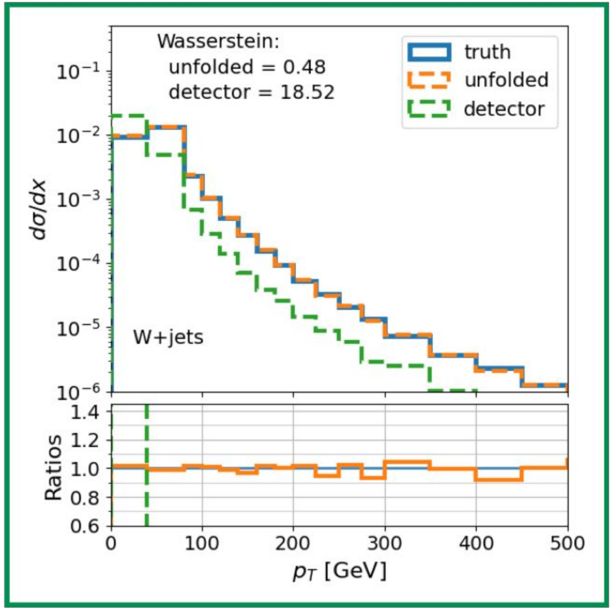




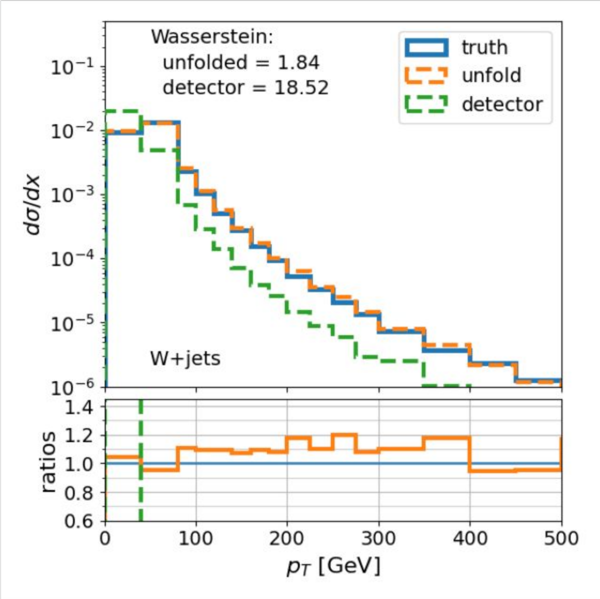
Unfolded results for $t\bar{t}$ (CT14lo, Vincia), which was not included in the training dataset, and comparison to other $t\bar{t}$ distributions that were in the training dataset. The ratios show the unfolded results divided by each of the listed $t\bar{t}$ distributions.

Additional Tests

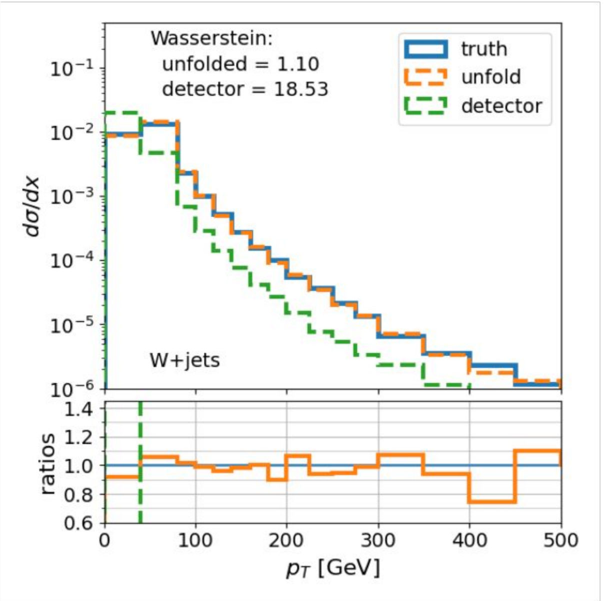
To investigate the impact of incorporating moments in the cDDPM's conditioning, we conducted experiments by training a cDDPM using datasets without including the moments and datasets where random numbers were assigned as the moments for the distributions.



Unfolder trained with multiple physics datasets, but without moments

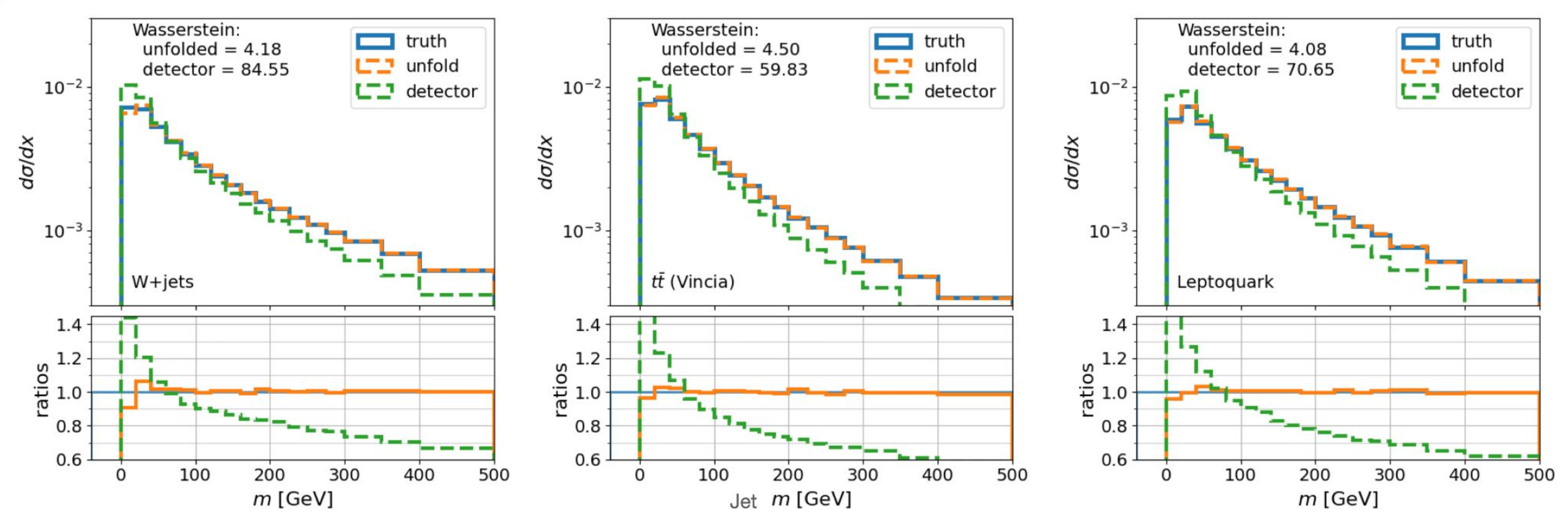


Unfolder trained with multiple physics datasets, with random moments



Maintaining Correlations

- Each object is defined as a vector of $[p_T, \eta, \phi, E, p_x, p_y, p_z]$
- Can we accurately reconstruct the mass from the unfolded quantities? → Yes!



Summary

- Using datasets that include the moments of the distributions, we can train a single cDDPM that can perform multidimensional object-wise unfolding on data from multiple physics processes, including those not seen during training.
- This unfolding maintains the correlations between the components of the object vector, allowing us to reconstruct other observables (like jet mass) from the unfolded results.

Future Work:

- Train cDDPM unfolders for other detector objects (leptons, MET) to reconstruct full events.
- Test with public datasets and compare performance to other unfolding approaches.
- Perform stress tests to find a failure case with physics data.
- Implement uncertainty estimation.

Open Questions:

- How can we optimize our selection of physics processes included in the training dataset?
- How many moments should be included for the best unfolding performance?

Thank you!

Paper:

https://arxiv.org/abs/2406.01507

Code: Coming soon!