Foundations of a Fast, Data-Driven, Machine-Learned Simulator

Jessica N. Howard
NSF Graduate Research Fellow
jnhoward@uci.edu

arXiv: 2101.08944

Jessica N. Howard[1], Stephan Mandt[2], Daniel Whiteson[1], Yibo Yang[2]

[1] Department of Physics and Astronomy, UC Irvine
[2] Department of Computer Science, UC Irvine

UCI University of California, Irvine

ML4Jets (hybrid)
July 6 - July 8, 2021

U.S. DEPARTMENT OF ENERGY | Office of Science

NSF

This work was supported in part by the following grants:
DE-SC0009920
DGE-1633631
DGE-1839285

Introduction

- Simulations are a crucial part of particle physics
- Current simulations (GEANT4) are accurate but computationally costly, and this is limiting discovery
- Hope Machine Learning (ML) can help make faster particle simulations

Previous Work

Substitute slowest parts of current simulations
or
Multiply simulated datasets

Augment
Current Simulations

Replace
Current Simulations

This Work

Taking the place of current simulations

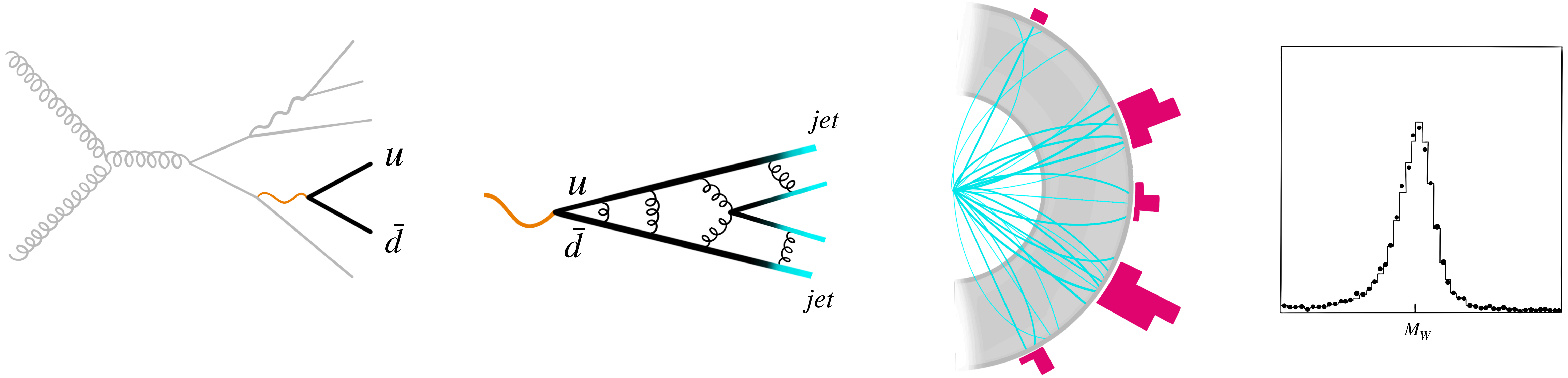
- Foundations of a method to **entirely replace** current simulations in analyses
  - Fast
  - Data-driven (trained using data from control regions)
  - Ability to build-in physics-motivated constraints
  - Easy to inspect and interpret

OTUS
Optimal **T**ransport based **U**nfolding and **S**imulation

Thinking about the problem

- Current simulations have 4 main stages that mimic real life

Parton Interactions	Showering	Detection	Reconstruction
$\mathcal{O}(10)$	$\mathcal{O}(100)$	$\mathcal{O}(10^6)$	$\mathcal{O}(10)$



Madgraph5 → **Pythia8** → **Delphes/GEANT4** →

Previous work has successfully produced **data from noise** at these various stages

[1] Butter, Anja, et al. arXiv: 1907.03764

[2] de Oliveira, Luke, et al. arXiv: 1701.05927
[3] Paganini, Michela, et al. arXiv: 1712.10321
[4] Buhmann, Erik, et al. arXiv: 2005.05334

[5] Otten, Sydney, et al. arXiv: 1901.00875
[6] Hashemi, Bobak, et al. arXiv: 1901.05282
[1] Butter, Anja, et al. arXiv: 1907.03764

OTUS →

Can we use ML to predict **reconstructed data** from **parton interactions** in a data-driven way?

Defining the Objective

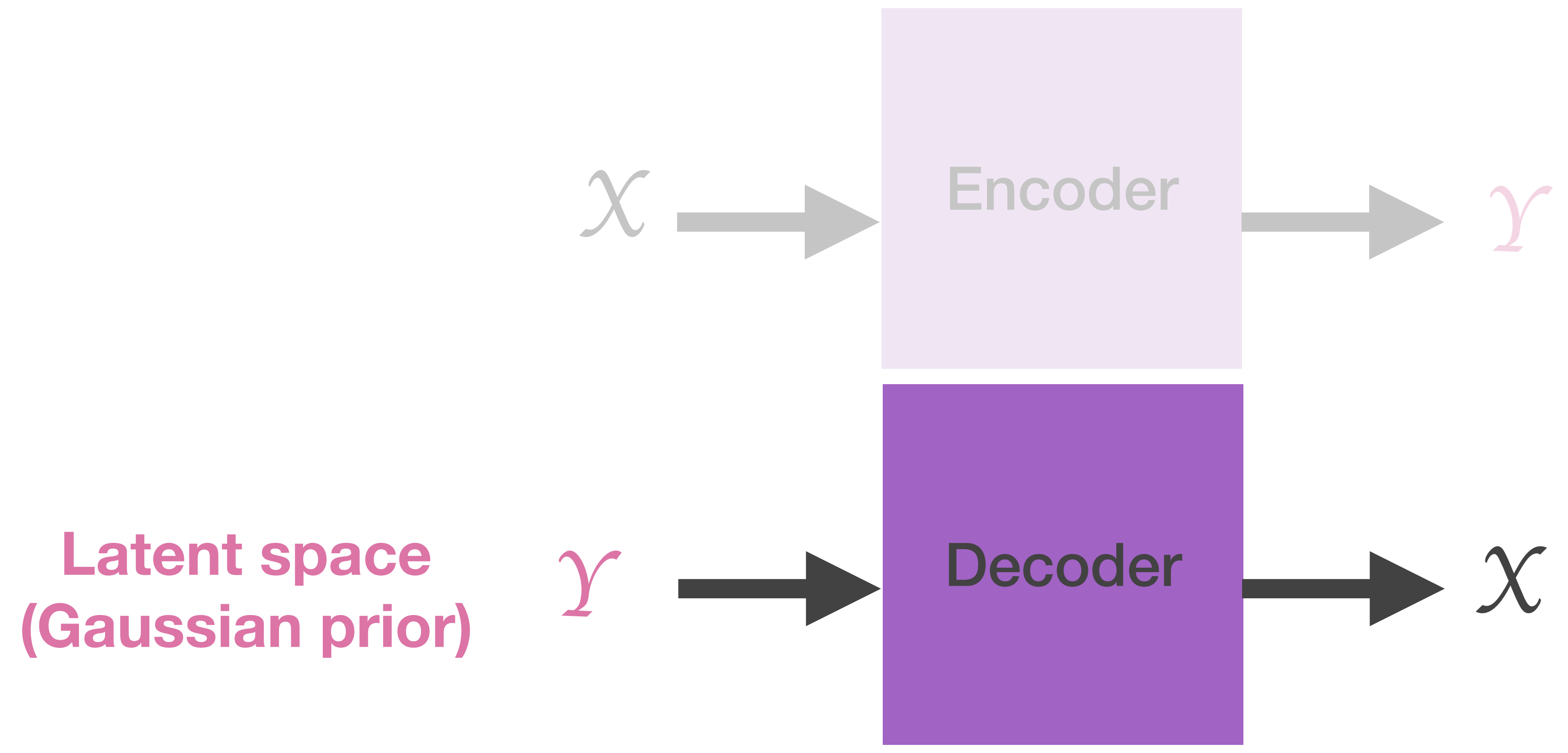
- What we want
  - Conditional mapping from **parton interactions** ($\mathcal{Z}$) to **observed data** ($\mathcal{X}$): $\mathcal{Z} \rightarrow \mathcal{X}$
  - Stochastic
  - The option to train on real data in control regions, not data produced by the current simulation

- Translating to ML terms
  - ***Conditional***, ***generative***, ***stochastic*** architectures trained via ***unsupervised learning***

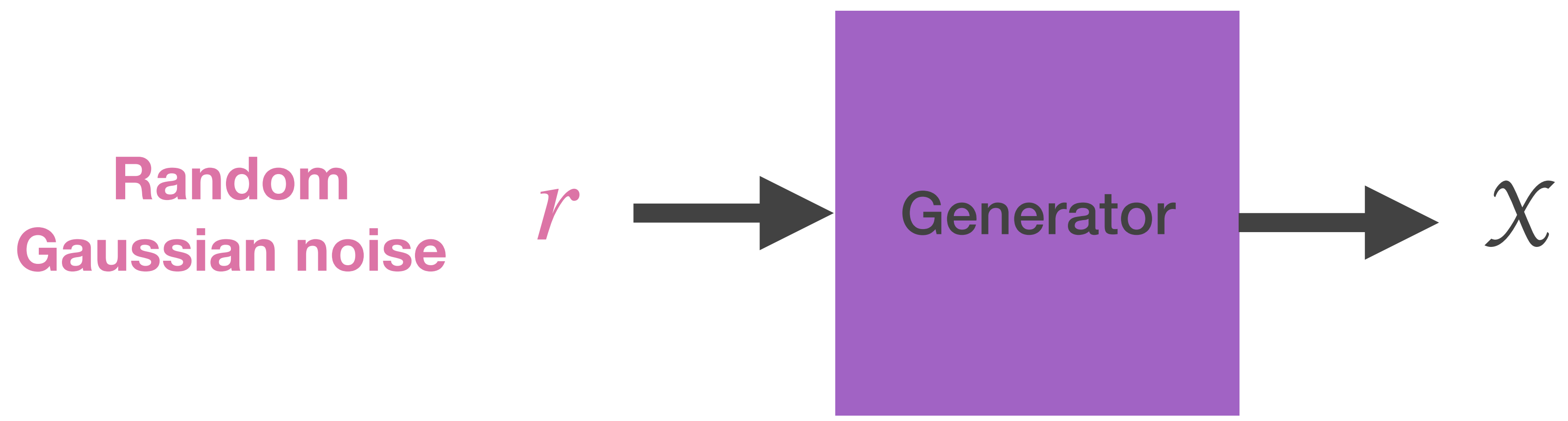
Rules

1. $\mathcal{Z} \rightarrow \mathcal{X}$
2. Stochastic
3. Unsupervised

Unsupervised, Generative ML



Variational Autoencoders (VAEs)
(arXiv: 1901.00875, 2005.05334)



Generative Adversarial Networks (GANs)
(arXiv: 1712.10321, 1701.05927, 1903.10563, 1901.00875)

Rules

1. $\mathcal{Z} \to \mathcal{X}$
2. Stochastic
3. Unsupervised

- Previous work has largely focused on GANs over VAEs
  - VAEs ~ GANs + extra optimization hurdle
- GANs are not ideal for this problem
  - They only mimic $\mathcal{X}$, they **do not** learn the transformation $\mathcal{Z}\to\mathcal{X}$
- What if we could replace r (or $\mathcal{Y}$) with $\mathcal{Z}$?
  - GANs: violates unsupervised learning tenant
  - VAEs: not immediately possible, but there may be a way out…

BONUS
If we succeed we also get a free "unfolding" map: $\mathcal{X} \to \mathcal{Z}$

Altering VAEs

- Traditional VAEs use KL-divergence as latent loss
  - Requires latent prior $p(y)$ to have tractable form (Gaussian)

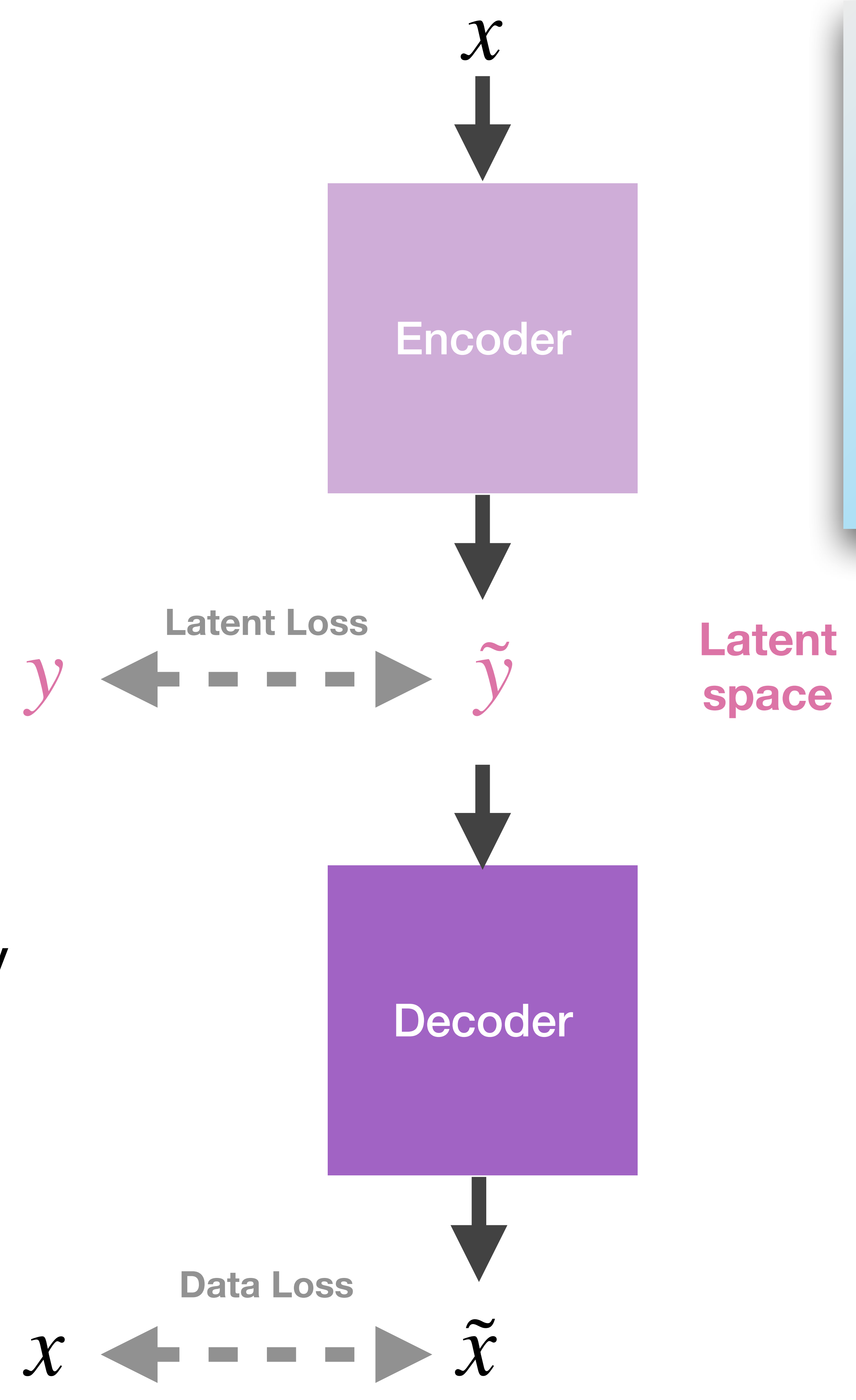
- The prior $p(z)$ of our theory space $(\mathcal{Z})$ often does not have a tractable form

- Is there a suitable replacement loss?
  - Answer: Yes!

- Recent paper: **Sliced Wasserstein Autoencoders (SWAE)**[1]
  - Loss based on Wasserstein distance from Optimal Transport theory

- What we get:
  - Latent prior can be **any sample-able distribution**
  - Allows for encoder and decoder to be **inherently stochastic**
  - Fixes other problems with VAEs along the way

[1] Kolouri, Soheil, et al. arXiv: 1804.01947 (see also Kolouri, Soheil, et al. arXiv: 1902.00434)



VAE Structure

Rules

1. $z \to x$ ✓
2. Stochastic ✓
3. Unsupervised ✓

Pulling it All Together

- Latent loss ($\mathcal{L}_{\text{Latent}}$)
  - SW distance for finite samples

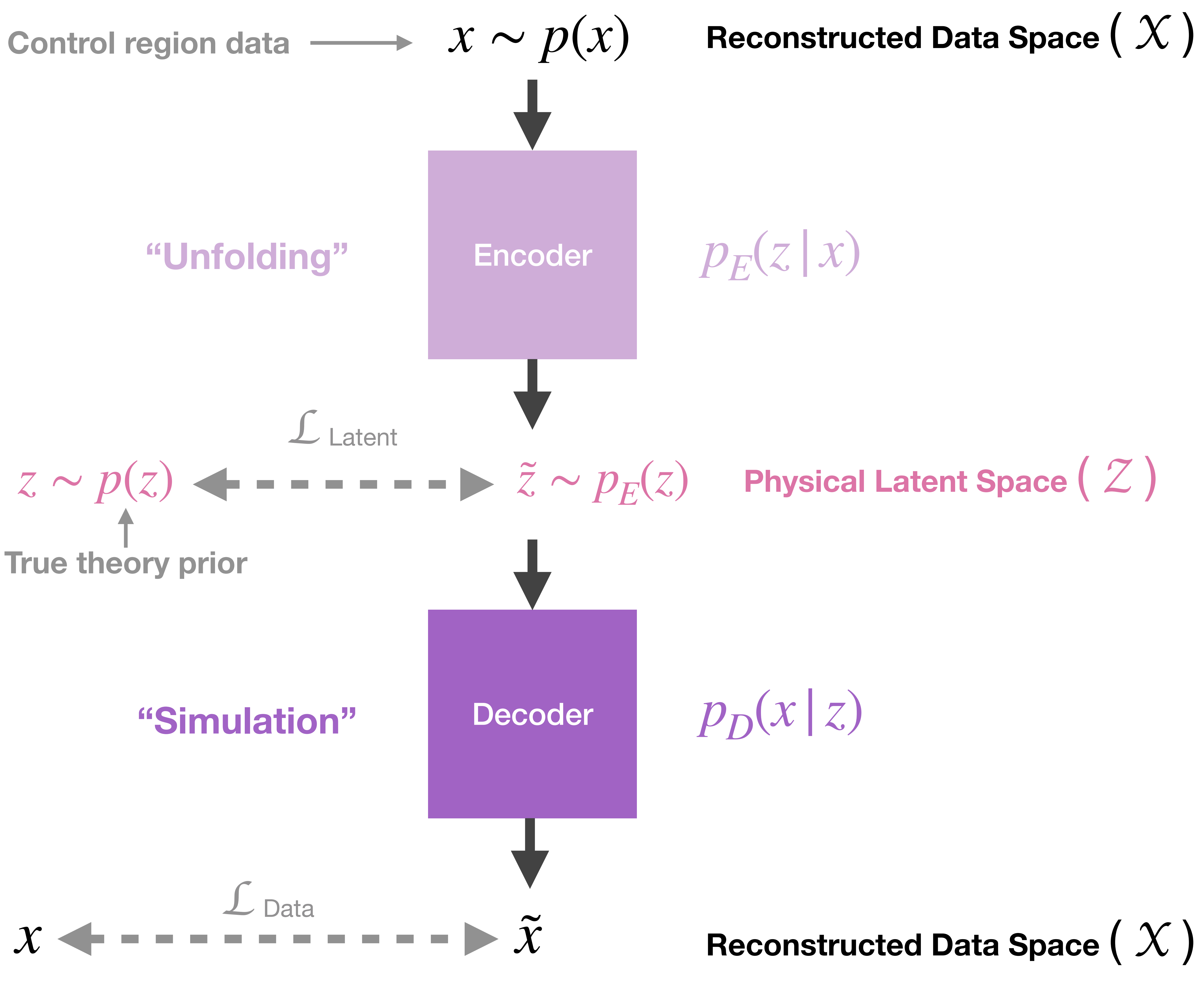
- Data loss ($\mathcal{L}_{\text{Data}}$):
  - Mean Squared Error (MSE)

- Total SWAE loss function:

$$\mathcal{L}_{\text{SWAE}} = \mathcal{L}_{\text{Data}} + \lambda \mathcal{L}_{\text{Latent}}$$

- Easy to add additional physically-motivated constraints

$$\mathcal{L} = \mathcal{L}_{\text{SWAE}} + \lambda_i' \mathcal{L}_i$$



SWAE Structure

Putting it to the test

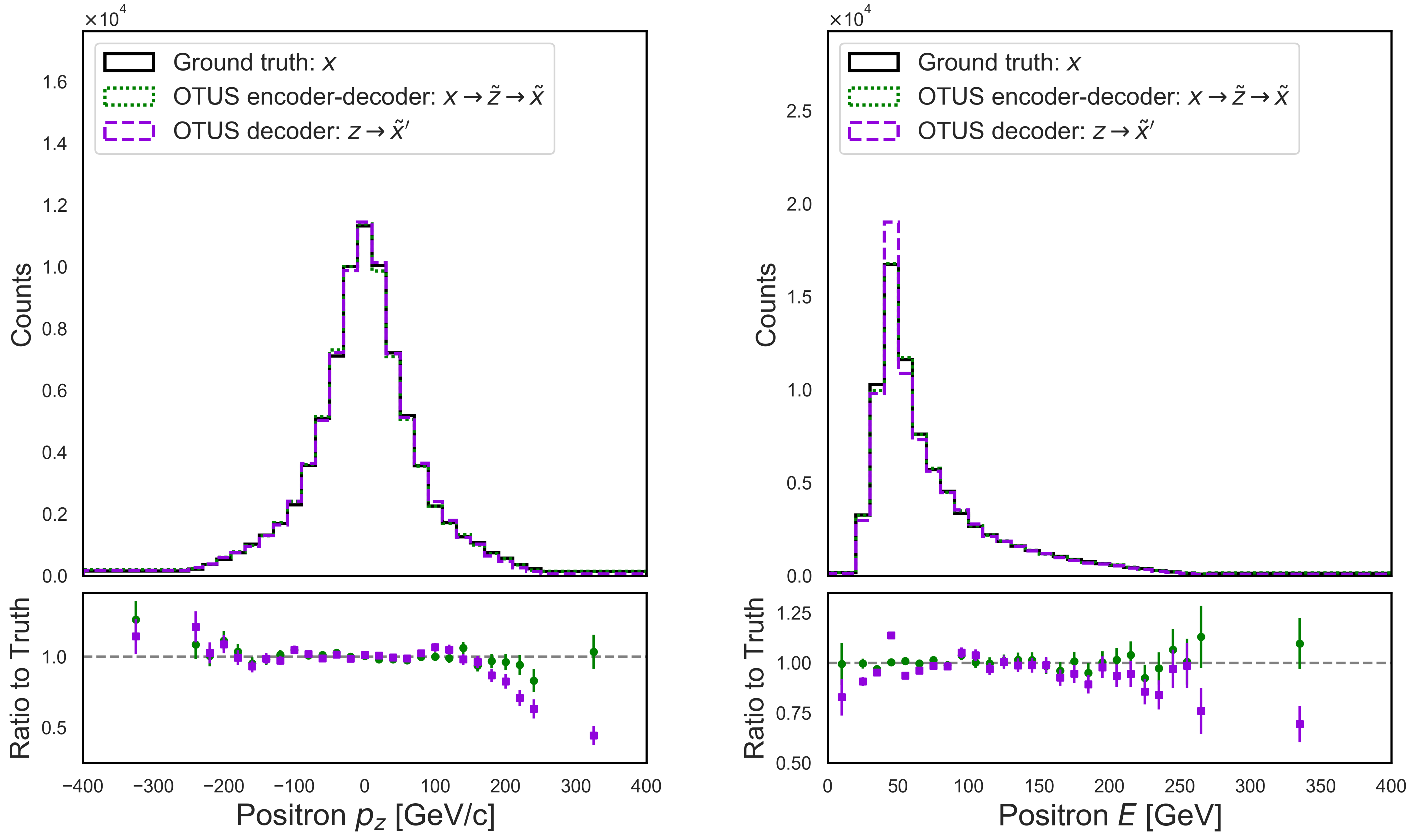
- **Test case 1:** $pp \to Z \to e^+e^-$
  - $\mathcal{Z}$ space: e^+ and e^- 4-momenta from Madgraph5 → [8 dimensions]
  - $\mathcal{X}$ space: e^+ and e^- 4-momenta from Delphes → [8 dimensions]

- **Test case 2:** $pp \to t\bar{t}$ (semileptonic)
  - $\mathcal{Z}$ space: e^-, $\bar{\nu}_e$, b, $\bar{b}$, u, and $\bar{d}$ 4-momenta from Madgraph5 → [24 dimensions]
  - $\mathcal{X}$ space: e^-, MET, and $jets$ 4-momenta from Delphes
    - Restrict to final state of exactly 4 jets → [24 dimensions]

- Note that $\mathcal{X}$ and $\mathcal{Z}$ **do not** need to have the same dimensions, but do in these tests

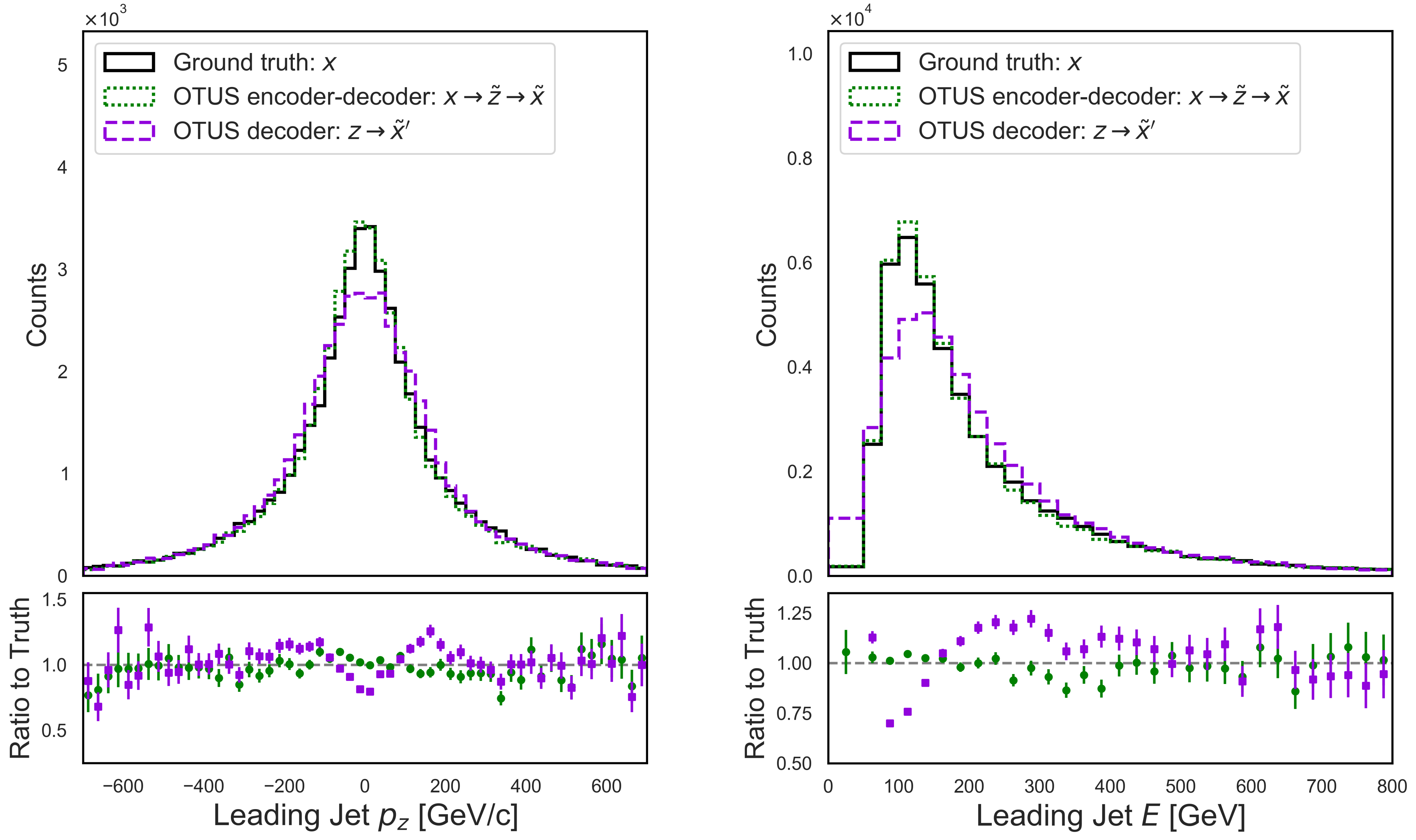
Simulation Results: Data Space ($\mathcal{X}$)

Positron (e^+)



$$pp \to Z \to e^+e^-$$

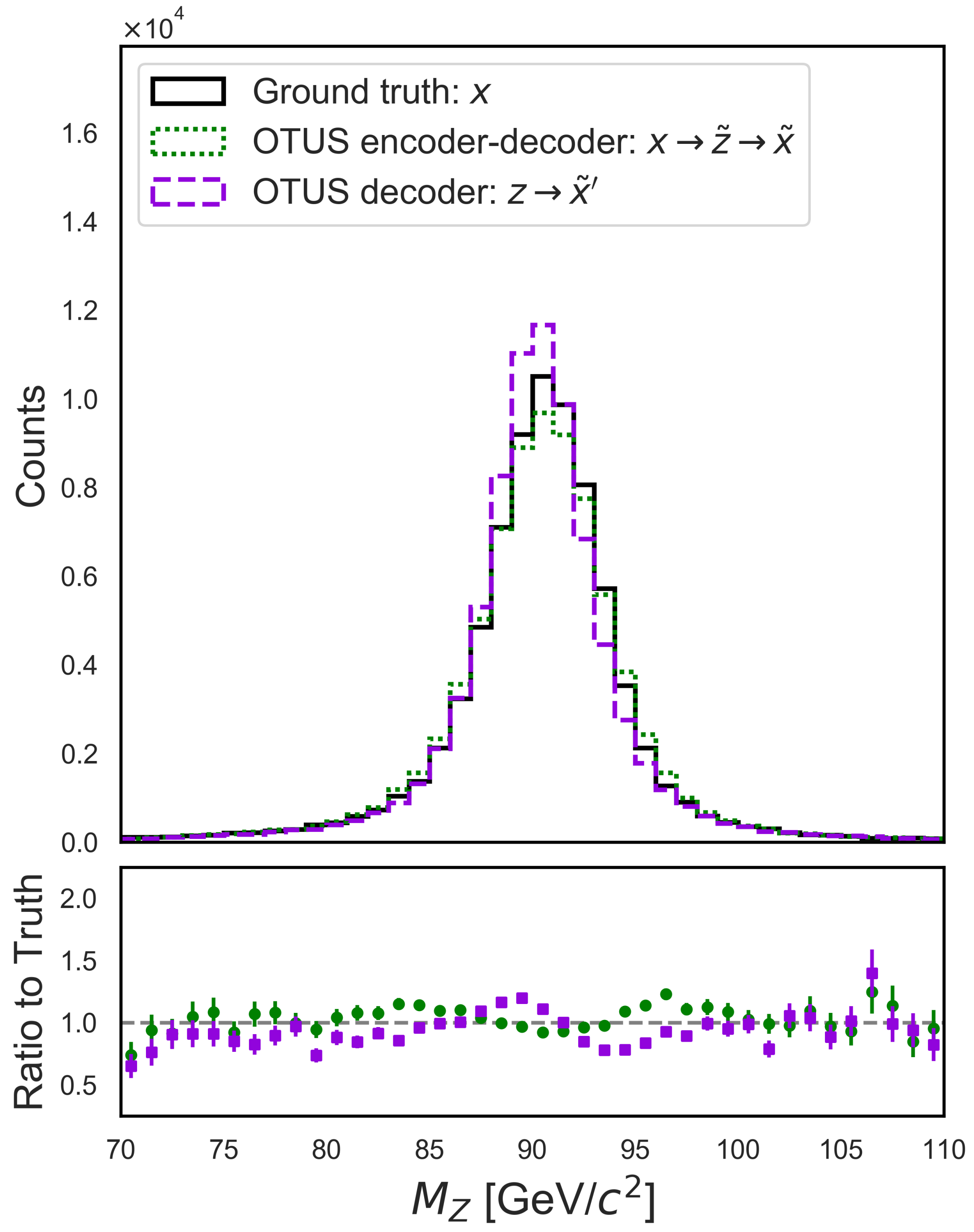
Leading jet

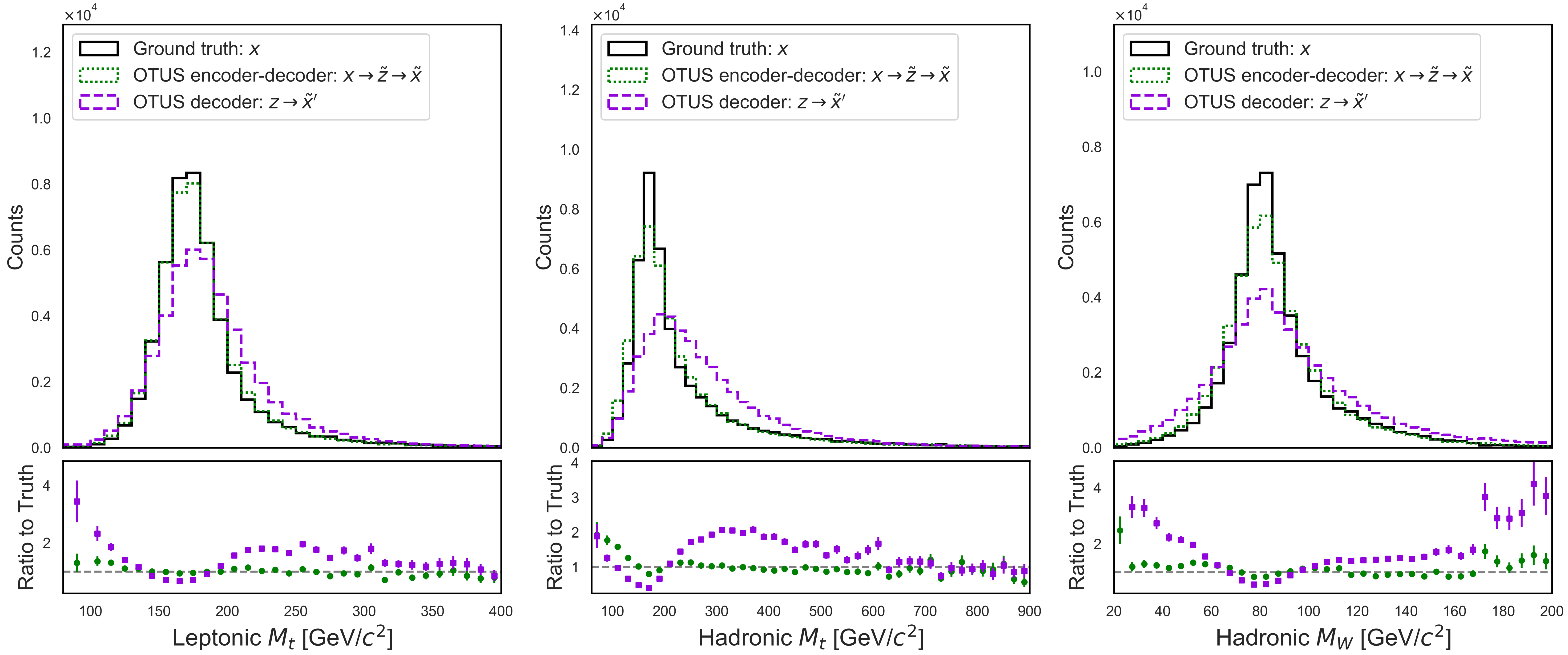


$$pp \to t\bar{t}$$

More Results: What did it learn?

- Not only interested in distribution matching, we care about the mapping being physical



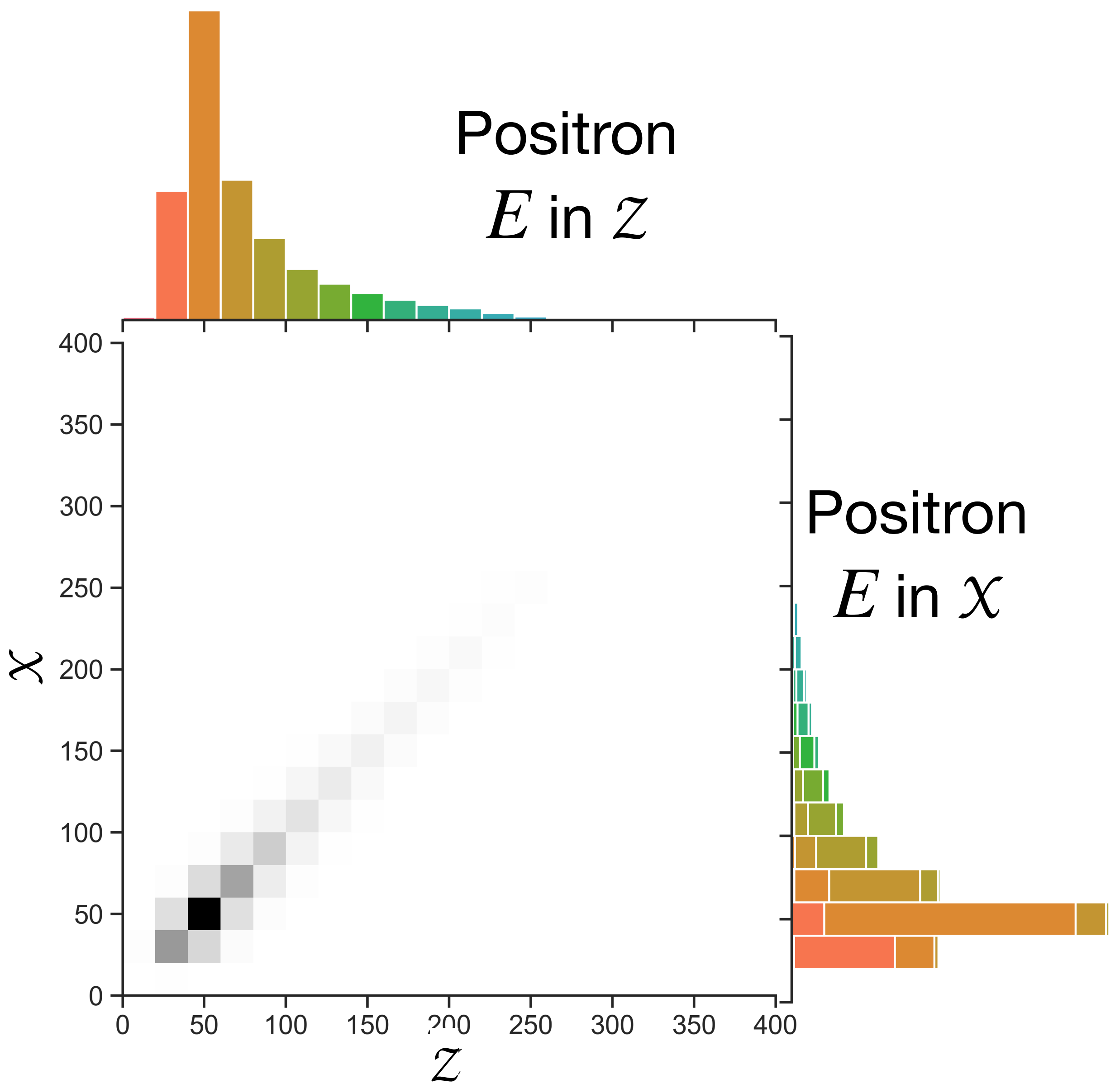


$pp \to Z \to e^+e^-$

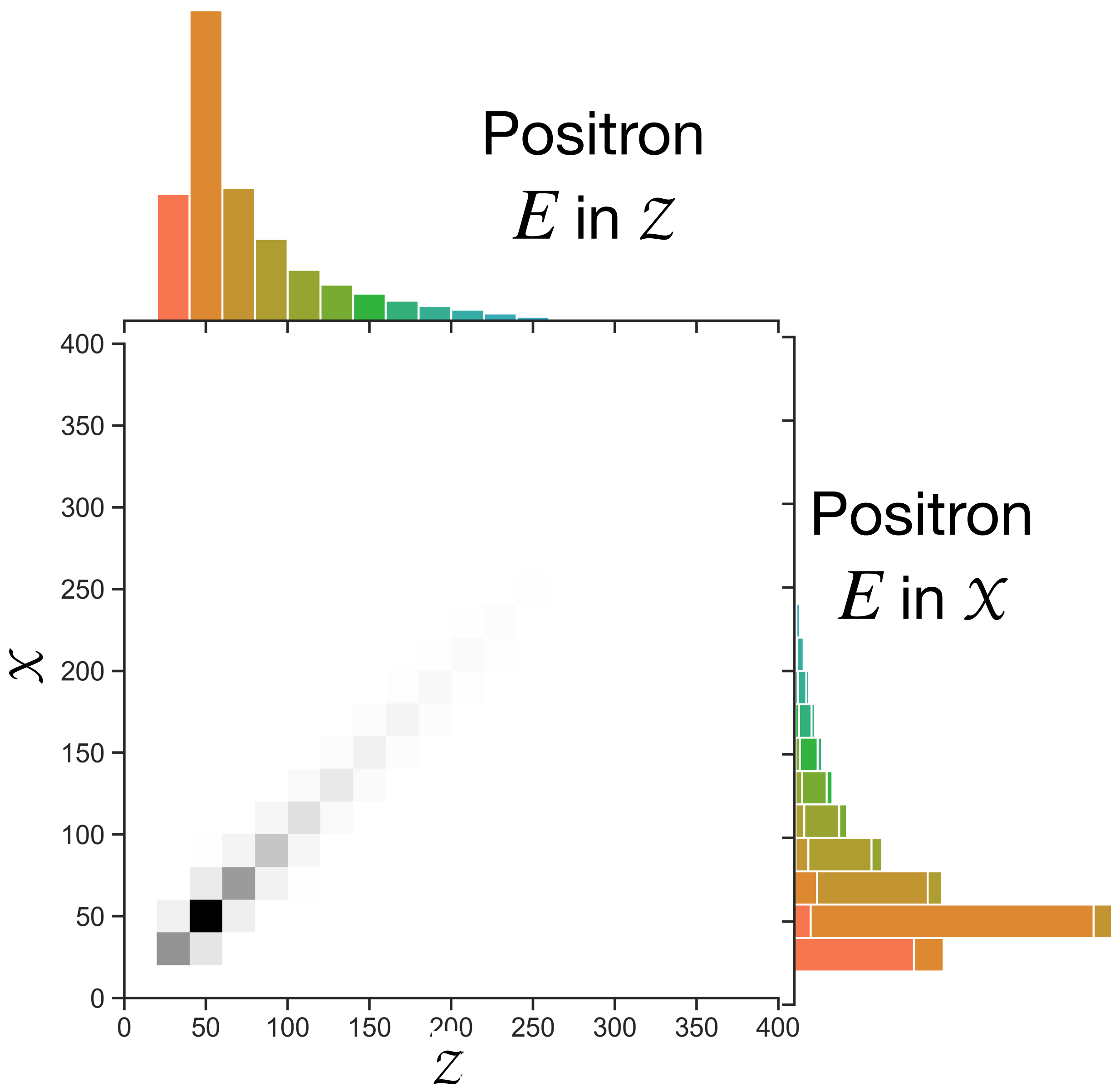
$pp \to t\bar{t}$

More Results: What did it learn?

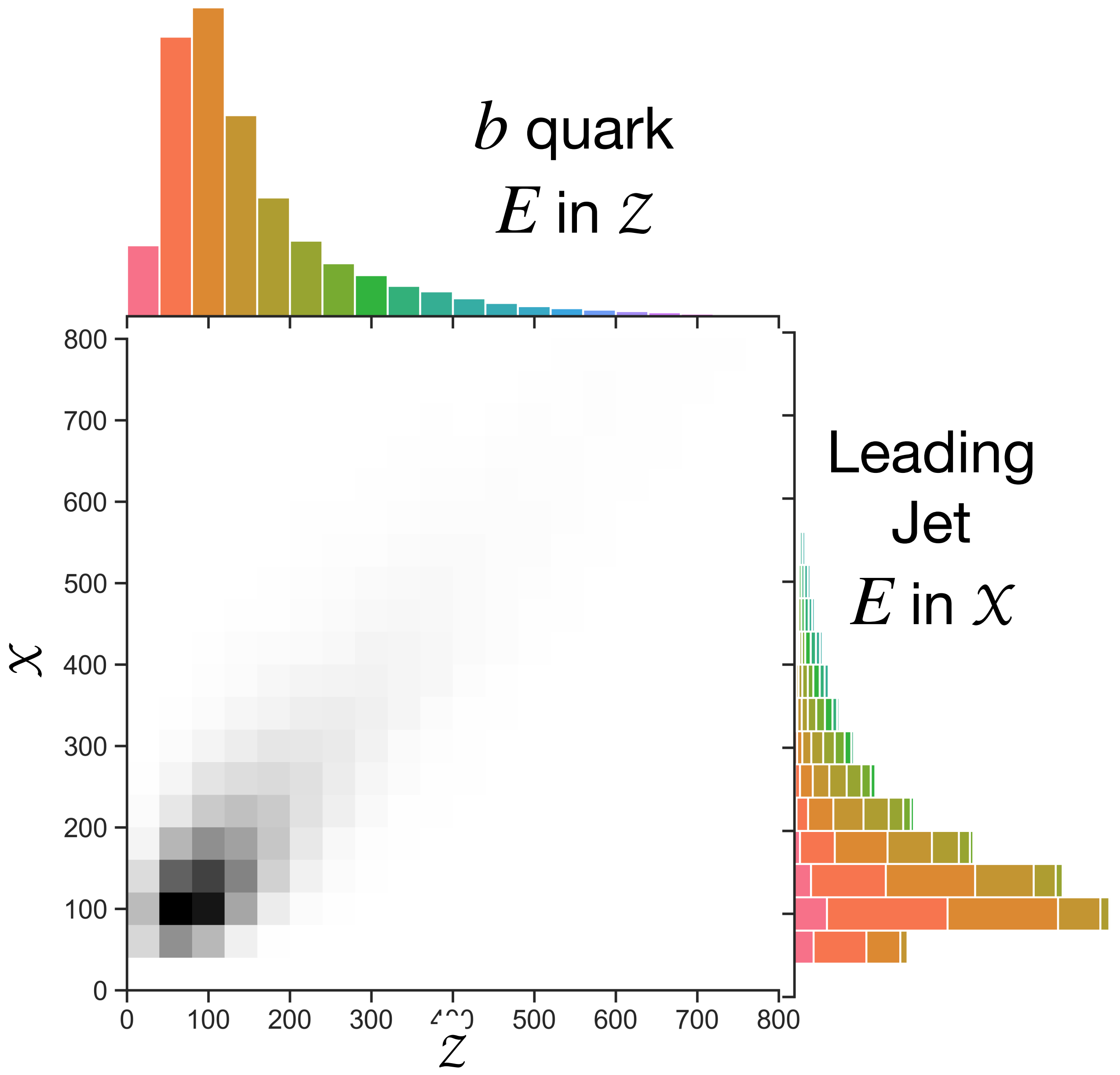
Transport plans:



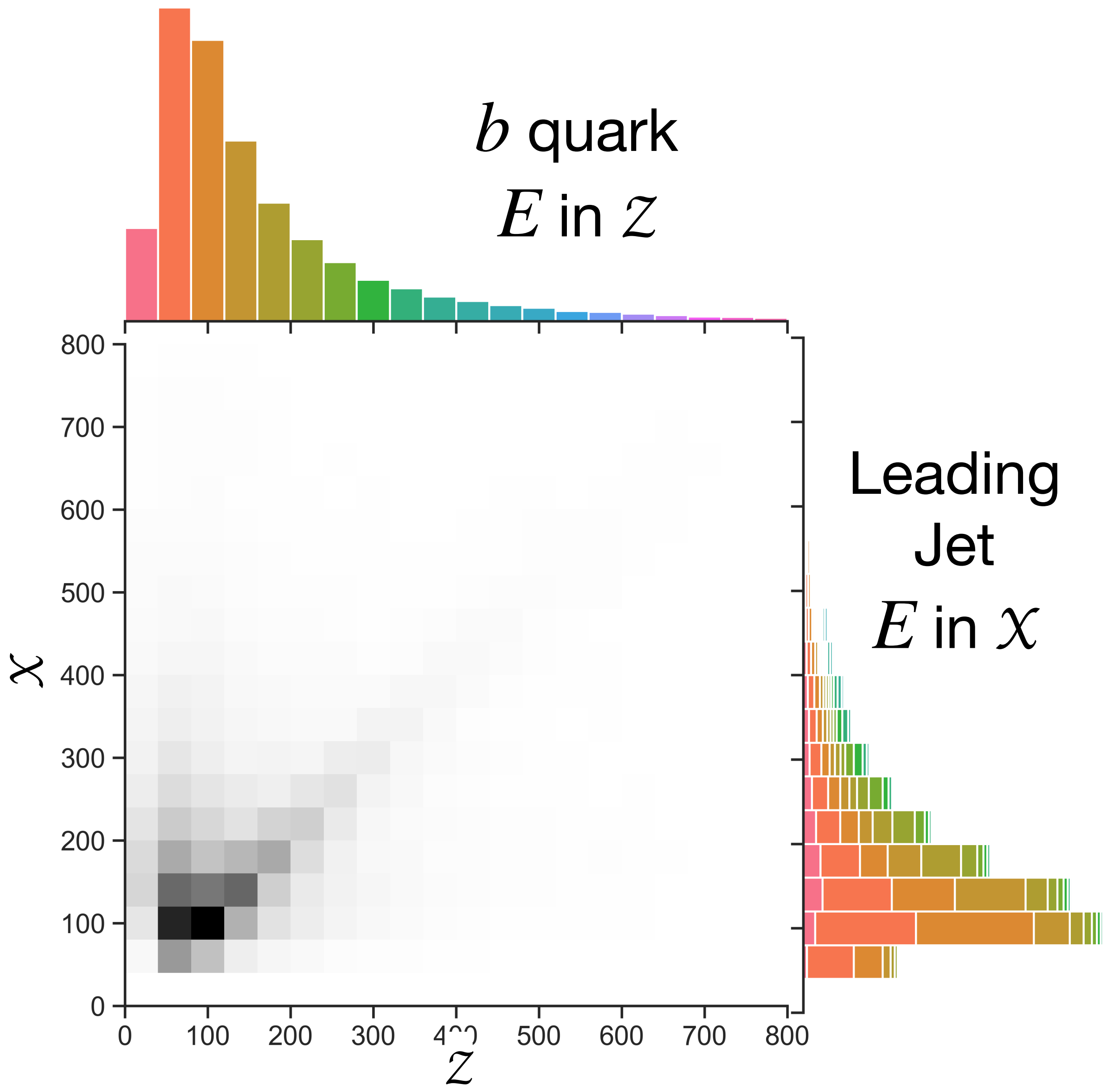
Decoder Transport plan ($\tilde{z} \to \tilde{x}$)



Delphes Transport plan ($z \to x$)



Decoder Transport plan ($\tilde{z} \to \tilde{x}$)

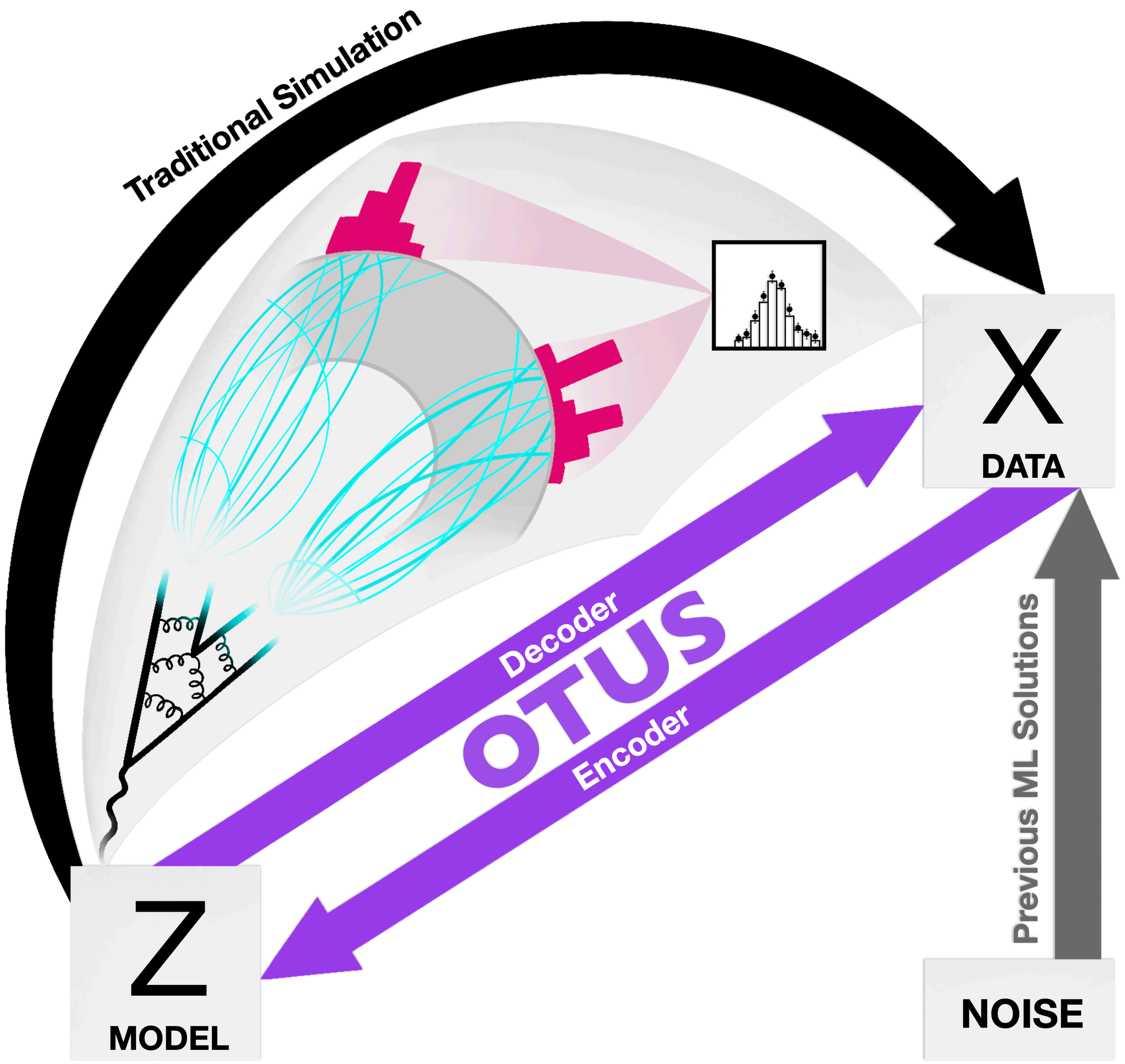


Delphes Transport plan ($z \to x$)

$$pp \to Z \to e^+e^-$$

$$pp \to t\bar{t}$$

Conclusion and Future Work



- New approach to fast simulation: Optimal Transport based Unfolding and Simulation (**OTUS**)
  - First step towards a data-driven, ML simulator
  - Bonus: Also get unfolding mapping
- Mathematically well-posed offering many advantages
- Performance on simple cases is promising but there is still work to do
- Future work
  - More robust description of the data
  - Handle variable particle types and numbers in data
  - Test ability to apply outside of control regions

Code → doi: 10.5281/zenodo.4706055

Data → doi: 10.7280/D1WQ3R

Backup Slides

(S)WAE Details

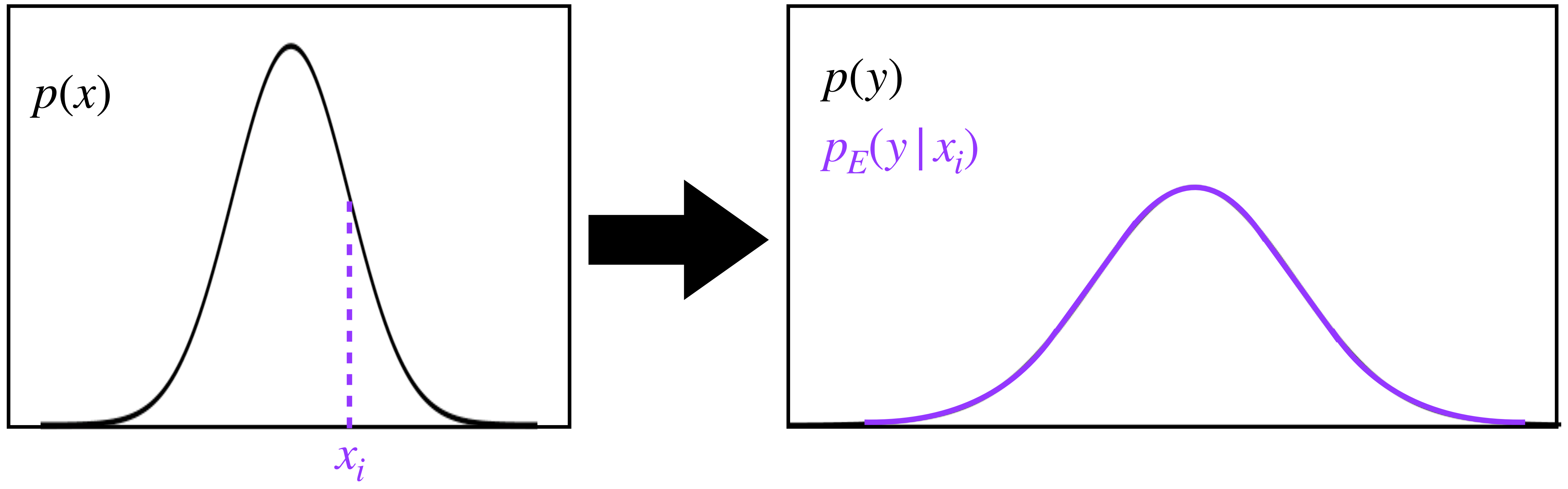
- **Wasserstein Autoencoders (WAE)**[1] first reimagined the VAE objective using Optimal Transport theory to solve another problem with traditional VAEs

Problem #1:
VAE latent loss encourages information collapse

$$\mathcal{L}_{\text{Latent}} = KL(\, p_E(y\,|\,x_i) \;||\; p(y))$$

Encoding distribution

Latent prior

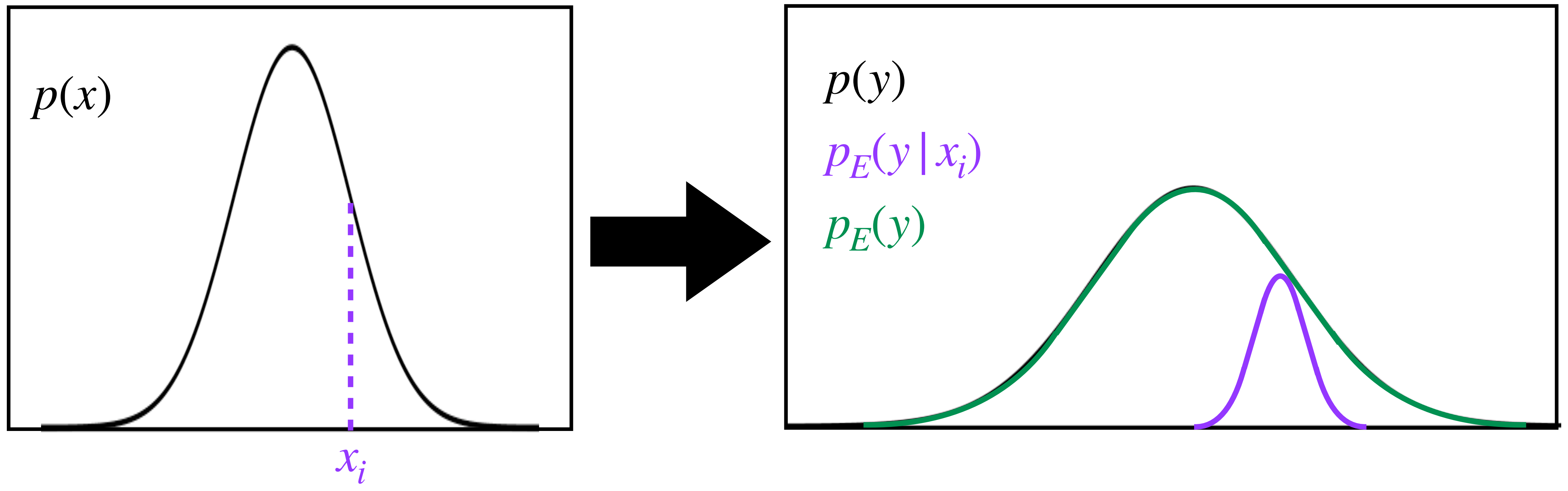


Every $x_i \sim p(x)$ is mapped to the whole $p(y)$ spoiling conditionality of y on x.

- **WAE** fixes this by matching distributions instead

$$p_E(y) = \int dx \; p_E(y\,|\,x)p(x) \quad \text{and} \quad p(y)$$

Marginalized encoding distribution



[1] Tolstikhin, Ilya, et al. arXiv: 1711.01558

(S)WAE Details

- **Wasserstein Autoencoders (WAE)**[1] loss function

$\mathcal{L}_{\text{Data}}$ $\mathcal{L}_{\text{Latent}}$

$$\mathcal{L}_{\text{WAE}} = \mathbb{E}_{x\sim p(x)}\mathbb{E}_{p_E(z|x)}\mathbb{E}_{\tilde{x}\sim p_D(x|z)}[c(x,\tilde{x})] + \lambda\, d_Z(p_E(y), p(y))$$

← **Theoretically arbitrary (incl. KL-divergence)**

Minimizing $\mathcal{L}_{\text{WAE}}$ minimizes the Wasserstein distance between $p_D(x)$ and $p(x)$

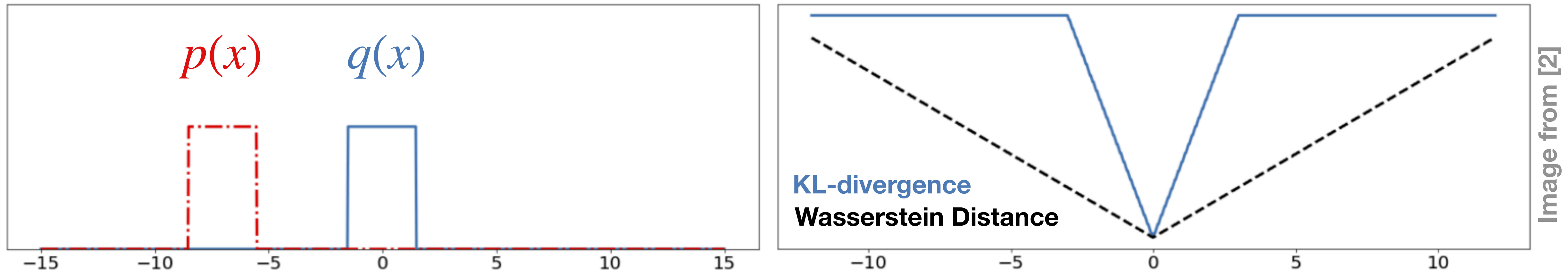
- **Sliced Wasserstein Autoencoders (SWAE)**[2] argue that d_Z should also be Optimal Transport based

[1] Tolstikhin, Ilya, et al. arXiv: 1711.01558 **[2] Kolouri, Soheil, et al. arXiv: 1804.01947**

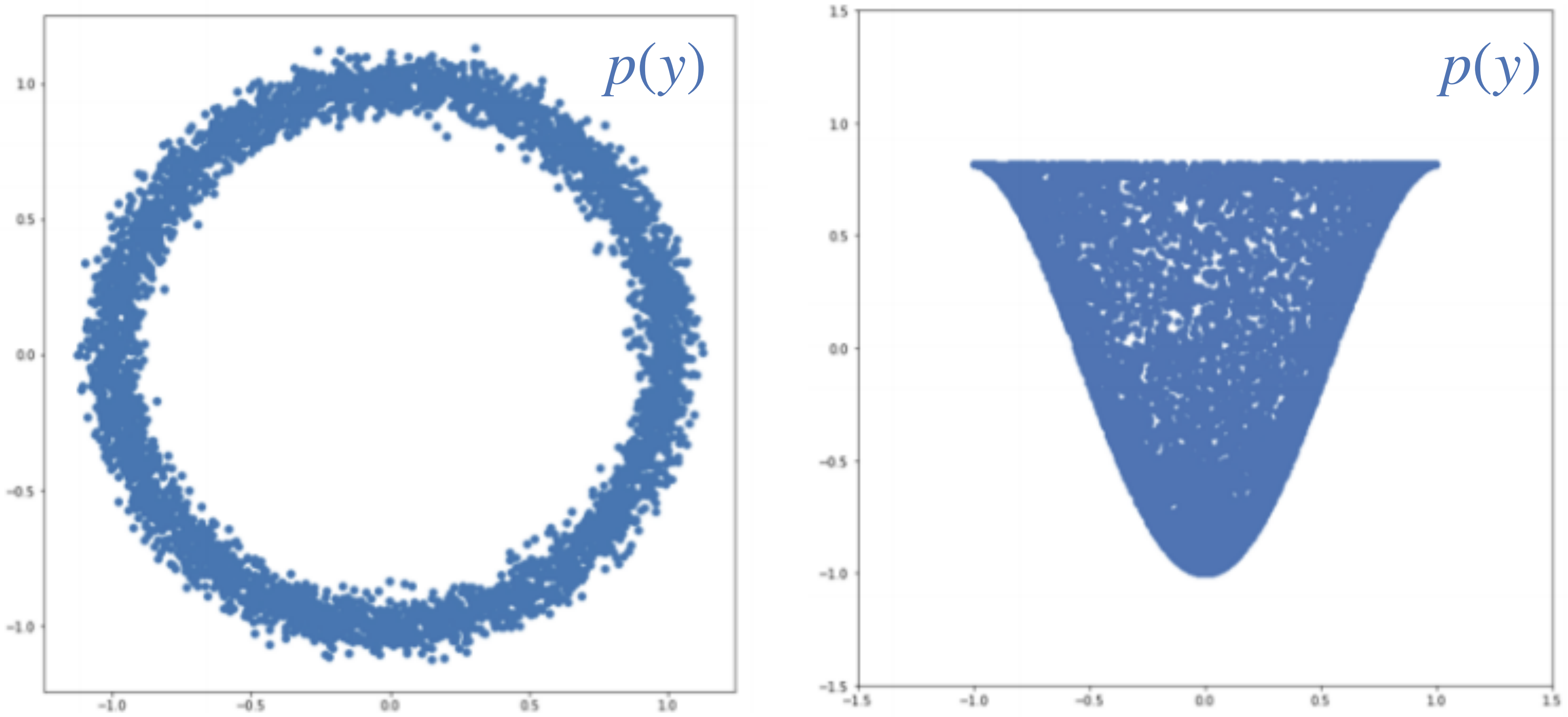
(S)WAE Details

- **Sliced Wasserstein Autoencoders (SWAE)[2]** wanted to solve additional problems arising from the use of KL-divergence or other similar cost functions

Problem #2:
KL-divergence is bad for non-overlapping distributions



Problem #3:
Inability to have an arbitrary latent-space priors which are only known from samples



Examples of possible latent-space priors

[1] Tolstikhin, Ilya, et al. arXiv: 1711.01558 [2] Kolouri, Soheil, et al. arXiv: 1804.01947

(S)WAE Details

- **Wasserstein Autoencoders (WAE)**[1] loss function

$$\overbrace{\mathcal{L}_{\text{WAE}} = \mathbb{E}_{x\sim p(x)}\mathbb{E}_{p_E(z|x)}\mathbb{E}_{\tilde{x}\sim p_D(x|z)}[c(x,\tilde{x})]}^{\mathcal{L}_{\text{Data}}} + \lambda \overbrace{d_Z(p_E(y), p(y))}^{\mathcal{L}_{\text{Latent}}}$$

← **Theoretically arbitrary (incl. KL-divergence)**

Minimizing $\mathcal{L}_{\text{WAE}}$ minimizes the Wasserstein distance between $p_D(x)$ and $p(x)$

- **Sliced Wasserstein Autoencoders (SWAE)**[2] argue that d_Z should also be Optimal Transport based
  - Choose d_Z to be the **Sliced Wasserstein Distance**

$$\mathcal{L}_{\text{SWAE}} = \mathbb{E}_{x\sim p(x)}\mathbb{E}_{p_E(z|x)}\mathbb{E}_{\tilde{x}\sim p_D(x|z)}[c(x,\tilde{x})] + \lambda\, d_{SW}(p_E(z), p(z))$$

- To understand why d_{SW} solves these problems we first need to understand the **Wasserstein Distance**, d_W

[1] Tolstikhin, Ilya, et al. arXiv: 1711.01558 [2] Kolouri, Soheil, et al. arXiv: 1804.01947

(S)W Distance Details

- **Wasserstein Distance**, d_W, measures the cost to morph one distribution into another via the optimal transport plan

- **Only tractable for univariate probability distributions**
  - Because the optimal transport plan is known

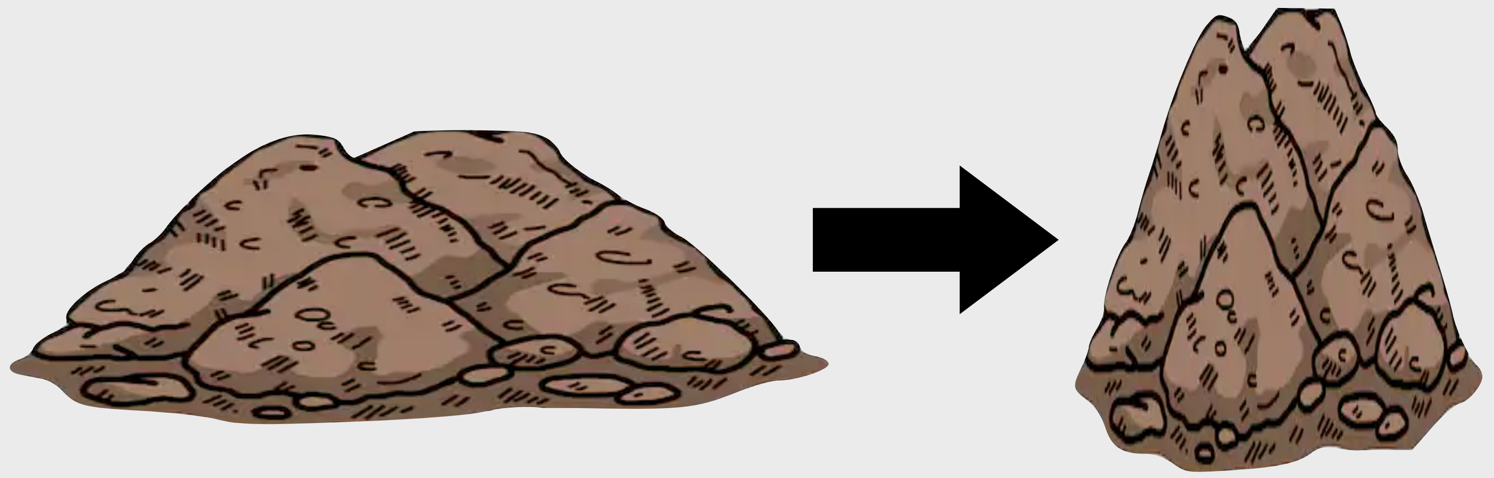
$$d_{W_1} = \int_0^1 dt\, | F^{-1}(t) - G^{-1}(t) |^{\alpha} \qquad \alpha \geq 1$$

- Where $F^{-1}(t)$ and $G^{-1}(t)$ are the inverse-CDFs of the distributions
  - These can be approximated from finite samples (EDF)

Wasserstein Distance (Earth Mover's Metric)

Given a pile of dirt

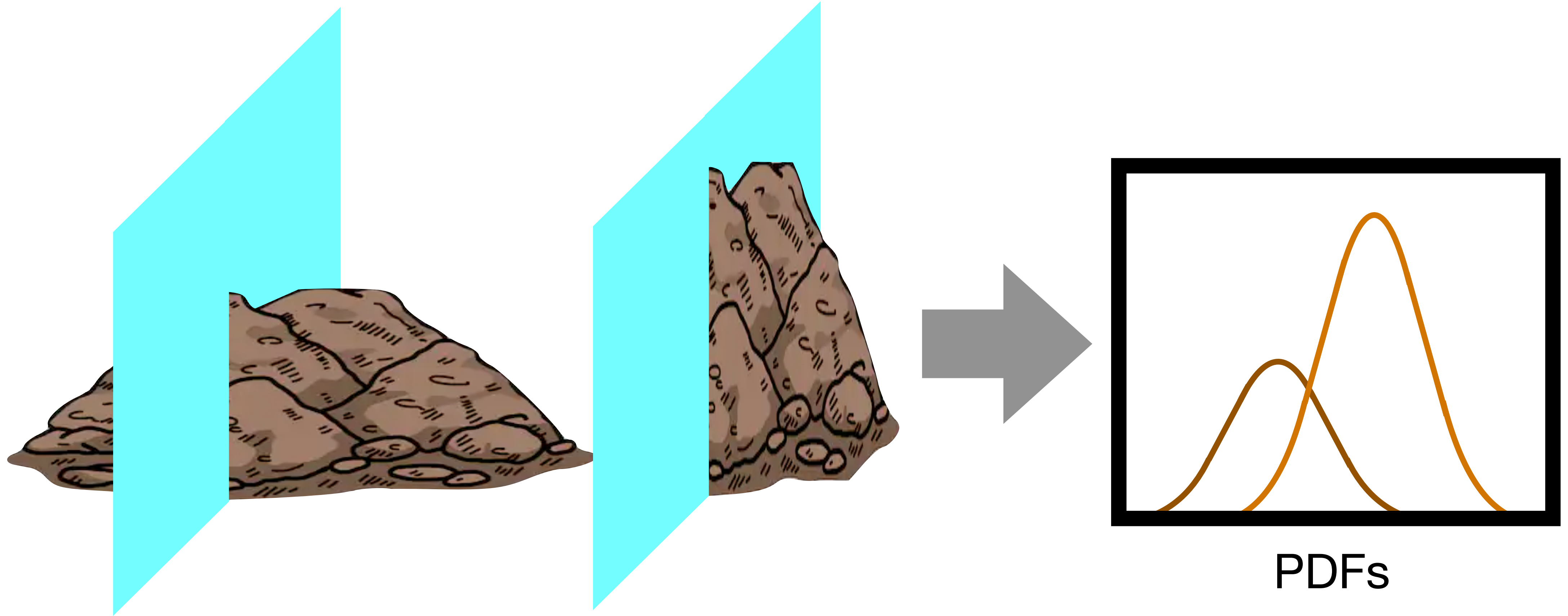
d_W is the cost of moving it some distance to form a pile with a different shape



According to the optimal transport plan

(S)W Distance Details

- **Sliced Wasserstein Distance**, d_{SW}, approximates the **Wasserstein Distance,** d_W, and is calculated by
  - Projecting multivariate distributions onto many 1D slices
  - Calculating d_{W_1} along each of those slices
  - Averaging the result
- In the limit of infinite slices, $d_{SW} = d_W$



"Slice" both distributions to get univariate PDFs

(S)W Distance Details

- How this solves the problems:

Problem #2:
KL-divergence is bad for non-overlapping distributions

d_{SW} is a true distance metric, unlike KL-divergence so it is always well-defined

Problem #3:
Inability to have an arbitrary latent-space priors which are only known from samples

We only need samples to calculate d_{SW}

$$\hat{d}_{SW} = \frac{1}{L * M} \sum_{l=1}^{L} \sum_{m=1}^{M} c((\theta_l \cdot z_m)_{sorted}, (\theta_l \cdot \tilde{z}_m)_{sorted})$$

M finite samples, L random slices

Test case 1: $pp \to Z \to e^+e^-$

- **Test case 1:** $pp \to Z \to e^+e^-$
  - $\mathcal{Z}$ space: e^+ and e^- 4-momenta from Madgraph5 → [8 dimensions]
  - $\mathcal{X}$ space: e^+ and e^- 4-momenta from Delphes → [8 dimensions]

Note that $\mathcal{X}$ and $\mathcal{Z}$ do not need to have the same dimensions, but do in these tests

- Additional constraints on mapping
  - We “anchor” the direction of the e^- momentum (fix the basis) with two additional losses

$$\mathcal{L}_{\text{A,E}} = \mathbb{E}_{x\sim p(x)}\mathbb{E}_{\tilde{z}\sim p_E(z|x)}[1 - \hat{p}^{e^-}_{x} \cdot \hat{p}^{e^-}_{\tilde{z}}] \qquad \mathcal{L}_{\text{A,D}} = \mathbb{E}_{z\sim p(z)}\mathbb{E}_{\tilde{x}'\sim p_D(x|z)}[1 - \hat{p}^{e^-}_{z} \cdot \hat{p}^{e^-}_{\tilde{x}'}]$$

- Additional constraints on properties of $\mathcal{Z}$ space and $\mathcal{X}$ space
  - Explicitly enforce the Minkowski metric as part of output of the network

Initially, the network picked up on this *implicit* relationship but its *explicit* inclusion improved the results

Test Case 2: $pp \to t\bar{t}$ (semileptonic)

- **Test case 2:** $pp \to t\bar{t}$ (semileptonic)
  - $\mathcal{Z}$ space: $e^-, \bar{\nu}_e, b, \bar{b}, u,$ and $\bar{d}$ 4-momenta from Madgraph5 → [24 dimensions]
  - $\mathcal{X}$ space: e^-, MET, and $jets$ 4-momenta from Delphes
    - Restrict to final state of exactly 4 jets → [24 dimensions]

- Technical difficulties introduced
  - Number of jets changes event to event
  - Quarks do not always map to the same jets (permutation)
  - Strict PT threshold for jets is difficult for a network to learn

General treatment requires a better way to structure the data

Imposed this explicitly

Dealing with the PT threshold

- Jets with PT< 20 GeV are excluded from our $\mathcal{X}$ space data ← **But this does not mean that there are no jets with PT < 20 GeV !**

- This threshold is not a property of the detector's transformation rather it is imposed on the data

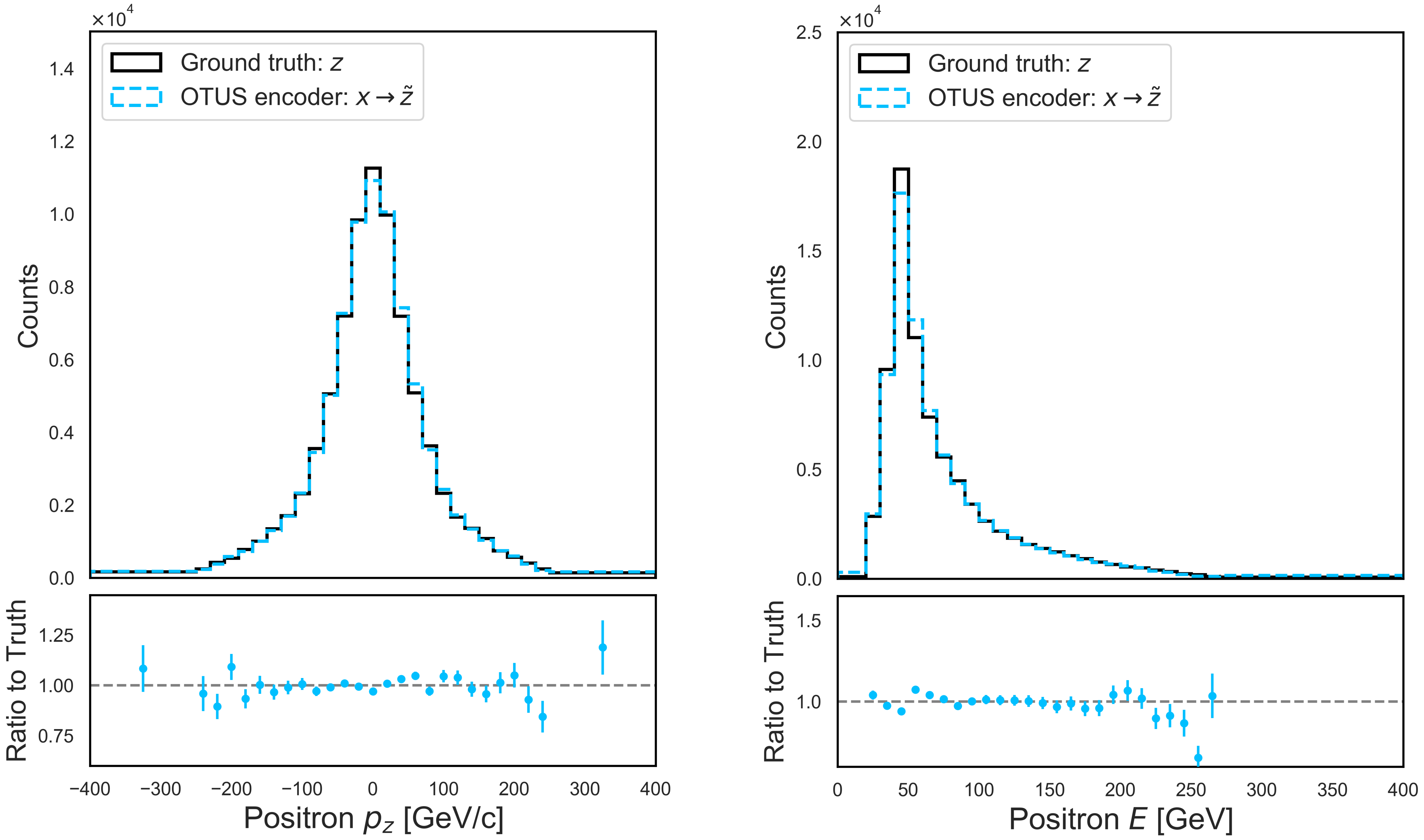
- If our goal is to learn the detector's transformation, we need a work around
  - Fortunately the math behind the modification is pretty straightforward
  - We only have access to a truncated version, $p(x)$, of the true $\mathcal{X}$ space data distribution, $p^*(x)$

$$p(x) \propto p^*(x) 1_S(x) \quad \text{where} \quad 1_S(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{otherwise} \end{cases}$$

- Practically this means
  - Only "passing" samples are used to calculate the data loss
  - Data loss term is weighted by the passing rate
  - **ResNet architecture was used for stable training → an initial identity bias** ← **Not ideal in all cases**

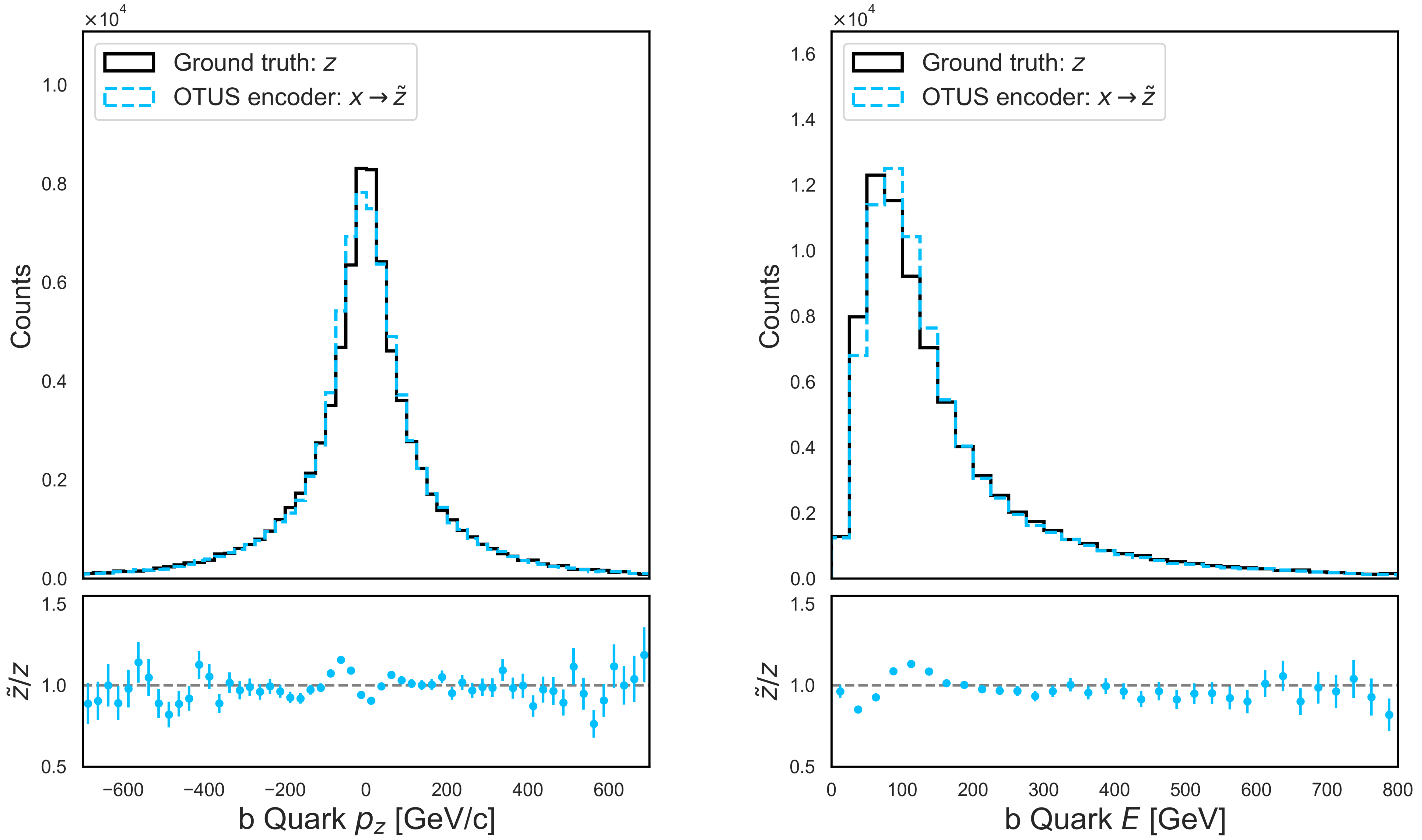
Unfolding Results: Latent Space ($\mathcal{Z}$)

Positron (e^+)



$pp \to Z \to e^+e^-$

b Quark



$pp \to t\bar{t}$

OTUS in other problems

- OTUS theoretically could be applied to any problem attempting to learn a transformation between arbitrary probability distributions

- Useful features:
  - Mappings can be deterministic or stochastic
  - The spaces can have the same or different dimensions
  - Distributions can have a known form or only be known from samples (d_{SW} vs $\hat{d}_{SW}$)
    - Note that using $\hat{d}_{SW}$ requires a large number of samples to accurately estimate the EDFs

- Other options:
  - Train only the decoder (or encoder) using $\hat{d}_{SW}$ as the loss (GAN alternative) with additional mapping constraints
  - Use a semi-supervised setup by substituting in an alternate estimation of d_W
    - If pairs {z, x} are known the transportation path is fixed resulting in an upper bound on d_W